INTERSPEECH 2017: Show & Tell Contribution
August 20–24, 2017, Stockholm, Sweden

Bob Speaks Kaldi

Milos Cernak, Alain Komaty, Amir Mohammadi, André Anjos, Sébastien Marcel

Idiap Research Institute, Switzerland
milos.cernak@idiap.ch

Abstract

This paper introduces and demonstrates Kaldi integration into Bob signal-processing and machine learning toolbox. The motivation for this integration is two-fold. Firstly, Bob benefits from using advanced speech processing tools developed in Kaldi. Secondly, Kaldi benefits from using complementary Bob modules, such as modulation-based VAD with an adaptive thresholding. In addition, Bob is designed as an open science tool, and this integration might offer to the Kaldi speech community a framework for better reproducibility of state-of-the-art research results.

Index Terms: Kaldi toolkit, Bob toolbox, speaker verification, reproducible research, open science

1. Introduction

Kaldi, a free and open-source toolkit, is designed for building ASR (Automatic Speech Recognition) systems [1]. Since 2011 when Kaldi was released, it attracted most of speech processing community, and it is seen as a modern way how to learn, build and collaborate on state-of-the-art ASR systems. Kaldi is designed rather as a R&D framework and it lacks a front-end designed to be easy to use for non-speech tasks. For example, although Kaldi contains state-of-the-art implementation of neural networks, their comparison with TensorFlow, Theano, or Caffe is not straightforward. It also uses an "open" protocol for data splits that makes more difficult to compare the results achieved on the same database by different research groups.

Bob, also a free and open-source signal-processing and machine learning toolbox [2], was released in about the same time as Kaldi. In addition to speech processing, Bob covers computer vision and video processing. It includes general machine learning and pattern recognition tools, such as dimensionality reduction, clustering, generative modelling, and discriminative classification. Bob is version controlled using GitLab, continually integrated (CI) and distributed on PyPI and Conda. Comparing to Kaldi, Bob is more general and includes a front-end for easy and fast running. Bob includes unified interfaces to more than 50 databases with fixed protocols for easy comparison of alternative algorithms, including for example NIST evaluation databases for speaker recognition and automatic speaker verification spoofing and countermeasures challenges.

The goal of this paper is to introduce and demonstrate Kaldi integration to Bob toolbox. This integration has mutual advantages. At one hand, Bob can benefit from using advanced speech processing tools developed in Kaldi, and at other hand, Kaldi tools can be efficiently extended with Bob and other python libraries for machine learning and speech signal processing. For example, Bob implements several voice activity detectors (VADs) that are missing in Kaldi, simple unsupervised energy-based VAD, modulation-based VAD with an adaptive thresholding, or easy use of the external or manually-labeled VAD. In addition, Bob is designed as the open research tool focused on reproducibility.

2. Case study on speaker verification

Bob is written in a mix of Python and C++ and contains a set of packages built using a common and uniform support. Each package contains a specific set of utilities to be used according to a specific application, such as `bob.db.*` for unified database interfaces, `bob.learn.*` for machine learning applications, `bob.bio.*` for running biometric recognition experiments, and so on. The overall block diagram of a Bob experiment is shown in Figure 1.

2.1. SPEAR: Speaker Recognition toolkit based on Bob

SPEAR is one of the Bob packages, providing a very simple, and researcher-friendly framework for executing speaker recognition experiments [3]. This is done by providing a large set of database interfaces, a number of preprocessors, feature extractors, and state-of-the-art modelling techniques such as GMM-UBM, inter-session variability (ISV), Joint Factor Analysis (JFA) and i-vectors[1]. All components can be run in parallel on a local machine or on a computation grid.

SPEAR is implemented as a derived package of `bob.bio.base`. The strength of SPEAR is that it profits from efficient C++ implementations and researcher-friendly Python. This approach allows for fast development of new features and add-ons.

2.2. Kaldi integration

Kaldi consists of C++ libraries and C++ executables that depend on some external libraries such as OpenFST and BLAS. Shell scripts, grouped mainly according to the training and evaluation data used, are provided as recipes.

To start integration, Kaldi is included as a new Bob dependency. All Bob dependencies are maintained as Conda packages. Conda is an open source, cross-platform, language-agnostic package and environment management system. The Conda Kaldi package is created as a binary Kaldi distribution, allowing one click installation with a simple command line. A tutorial is provided on installation and usage of the Kaldi integration package[2].

Figure 2 shows how functionally of Kaldi is integrated into existing SPEAR package. Kaldi integration is done by re-using binaries wrapped around a python friendly API (functions). For example, feature extraction wrapper for speaker verification includes computation of standard MFCC or PLP features, adding delta features, and applying sliding CMVN. Another GMM python wrapper is created for UBM training with diagonal or full-covariance GMM models. In addition to Kaldi implementation, MAP adaptation of diagonal GMM model was also wrapped. I-vector and PLDA scoring is implemented in another python wrapper module, and so on.

[1] `https://pythonhosted.org/bob.bio.spear`
[2] `https://pypi.python.org/pypi/bob.kaldi`

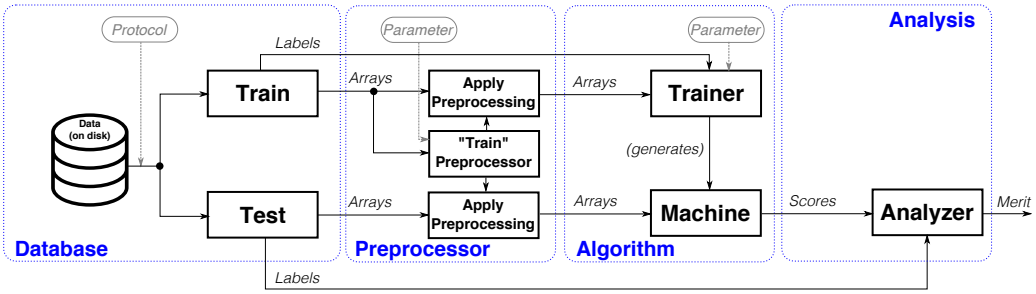

Figure 1: *A flow chart of the general processing chain in Bob. It is organised in a pretty general manner to fit a great number of problems and Machine Learning and Pattern Recognition with minor modifications.*

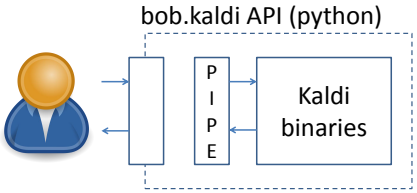

Figure 2: *A python wrapper for Kaldi binaries. API calls use just numpy arrays for data flow with other modules.*

Table 1 shows sample results of native Bob and Kaldi speaker verification algorithms on the Mobio male database [4].

Table 1: *Native Bob and Kaldi speaker verification based on MAP adaptation of GMM-UBM models with 512 Gaussians, using Bob and Kaldi defaults. Both implementations yield similar performance, despite tuned slightly differently.*

System	EER	HTER
Native Bob	23.41%	12.06%
Kaldi Bob	17.51%	12.44%

3. Current and future work

Current work is focused on integration of the Kaldi speaker recognition recipes to Bob. This will naturally evolve into integration of the Kaldi speech recognition recipes. ASR training recipes become new Trainers (see Figure 1), ASR decoders new Machines, and WER scoring new Analyzers.

Simultaneously with new Kaldi functionality, we start to port current workflows for speech processing to the BEAT platform [5] (see a screenshot on Figure 3). The BEAT platform is a European computing e-infrastructure for Open Science proposing a solution for open access, scientific information sharing and re-use including data and source code while protecting privacy and confidentiality. It allows easy online access to experimentation and testing in computational science. Data from different experiments can be easily compared and searched. The platform also provides an attestation mechanism for your reports (scientific papers, technical documents or certifications).

4. Acknowledgements

The research leading to this work has received funding from the Swiss Center for Biometrics Research and Testing, the ANR-SNSF ODESSA project and the Research Council of Norway SWAN project.

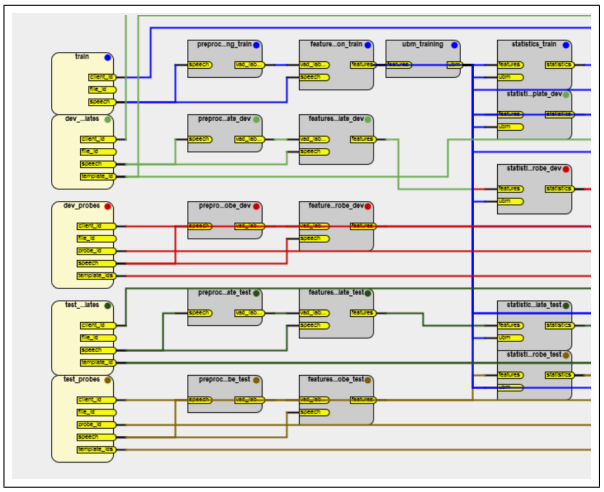

Figure 3: *A screenshot of the speaker verification toolchain in the BEAT platform, showing interconnected data and algorithmic modules. The platform provides real-time feedback on the data flow. For example, the green colour represents successfully processed processing paths.*

5. References

[1] 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 *Proc. of ASRU*. IEEE SPS, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[2] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: a free signal processing and machine learning toolbox for researchers," in *20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*. ACM Press, Oct. 2012.

[3] E. Khoury, L. El Shafey, and S. Marcel, "Spear: An open source toolbox for speaker recognition based on Bob," in *Proc. of ICASSP*, 2014.

[4] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernockỳ, N. Poh, J. Kittler, A. Larcher, C. Levy *et al.*, "Bi-modal person recognition on a mobile phone: using mobile phone data," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 635–640.

[5] A. Anjos, L. El-Shafey, and S. Marcel, "BEAT: An Open-Source Web-Based Open-Science Platform," *ArXiv e-prints*, Apr. 2017. [Online]. Available: https://arxiv.org/abs/1704.02319