IDIAP RESEARCH REPORT

IMPLEMENTATION OF THE STANDARD I-VECTOR SYSTEM FOR THE KALDI SPEECH RECOGNITION TOOLKIT

Srikanth Madikeri Subhadeep Dey Petr Motlicek
Marc Ferras

Idiap-RR-26-2016

OCTOBER 2016

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Implementation of the Standard I-vector System for the Kaldi Speech Recognition Toolkit

Srikanth R. Madikeri, Subhadeep Dey, Petr Motlicek and Marc Ferras
Idiap Research Institute, Switzerland
email: {msrikanth,sdey,motlicek, mferras}@idiap.ch

***Abstract*—This report describes implementation of the standard i-vector-PLDA framework for the Kaldi speech recognition toolkit. The current existing speaker recognition system implementation is based on the Subspace Gaussian Mixture Model (SGMM) technique although it shares many similarities with the standard implementation. In our implementation, we modified the code so that it mimics the standard algorithms in the i-vector based speaker recognition system. The implementation is compared with the existing Kaldi recipe for speaker recognition on the NIST SRE 2008 evaluation set. The entire implementation is made available at https://github.com/idiap/kaldi-ivector under the Apache 2.0 license.**

***Index Terms*—Speaker Recognition, Formant emphasis, i-vector**

I. Introduction

State-of-the-art speaker recognition systems build speaker models based on Mel Frequency Cepstral Co-efficients (MFCC) feature vectors extracted from speech utterances. The sequence of feature vectors from an audio sample are adapted with respect to a prior model such as the Universal Background Model (UBM) to form a vector of stacked means called the supervector [1]. The supervectors are transformed to a low-dimensional representation of the speech utterance with Total Variability Space (TVS)-based modelling. This low-dimensional representation is called the i-vector (*identity* vector). The i-vectors are further processed using discriminative modelling techniques (eg: LDA [2, 3] and PLDA [4]) to re-orient the TVS and get rid of the channel effects at the model level. This constitutes a typical speaker recognition system.

This framework is implemented for the Kaldi toolkit so that it can be seamlessly used with other speech technologies available in there. The rest of the document is organized as follows. First, the standard i-vector systems is described in Section II. Next, the standard scoring strategies are described in Section III. The organization of the package is discussed in Section IV. In Section V the results on the NIST SRE 2008 dataset is presented. Finally, Section VI summarizes the results.

II. Standard i-vector system

An i-vector is a fixed dimensional representation of a speech recording. Given a supervector $\mathbf{s}$ obtained from a speech recording with respect to a UBM/GMM (Universal Background Model/Gaussian Mixture Model) with a mean supervector $\mathbf{m}$, the total variability space model is given by

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

where $\mathbf{T}$ is the matrix defining the low-dimensional space and $\mathbf{w}$ is the corresponding i-vector for $\mathbf{s}$.

If there are C components in the UBM and the feature vectors obtained from the recording are F-dimensional, the supervector has CF elements. The i-vector dimensions R is typically much less than CF.

The UBM here is represented by a Gaussian Mixture Model (GMM) (Section 2.3.9 in [5]). Let the weights of the c^{th} mixture of the GMM be ω_c. Thus, the density function of the GMM represented by parameter set $\boldsymbol{\theta}$ for a feature vector $\mathbf{x}$ is given as

$$P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{c=1}^{C} \omega_c \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \tag{2}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ represents the value of probability density function of a Gaussian distribution with mean $\boldsymbol{\mu}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$ (see Section 2.3 in [5]). Note that $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean and covariance parameters of the c^{th} component of the GMM.

Given a speech recording $\mathbf{O}$ that contains a sequence of feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$, the i-vector is computed from the sufficient statistics of the model in Equation 1. The sufficient statistics include the zeroth and first order statistics. The centered first order statistics are computed as follows

$$\mathbf{f}_c = \left(\sum_t \gamma_{t,c} \mathbf{x}_t - \eta_c \boldsymbol{\mu}_c \right), \tag{3}$$

where $\mathbf{f}_c$ is the c^{th} $F \times 1$ block of the supervector such that

$$\mathbf{f} = [\mathbf{f}_1^t \mathbf{f}_2^t ... \mathbf{f}_C^t]^t,$$

$\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean vector and covariance matrix of the c^{th} mixture of the UBM, respectively, and $\gamma_{t,c}$ is defined as follows

$$\gamma_{t,c} = \omega_c \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c). \tag{4}$$

In Equation 3, η_c is the effective number of feature vectors aligned with the c^{th} mixture. The vector $[\eta_1 \eta_2 \cdots \eta_C]^t$ represents the zeroth order statistics for the utterance. The alignment of the feature vectors with respect to a mixture is

Corresponding author contact: Idiap Research Institute, Martigny, Switzerland
email: msrikanth@idiap.ch

soft. Only the top scoring components are retained and the posteriors are re-normalized to sum to 1.0. In our systems, the top 20 scoring posteriors are retained and re-normalized for every feature vector. In Equation 3 we avoid the whitening of the statistics with respect to the UBM means and variances, as it is often the case in conventional i-vector system implementations [6, 7].

The hyperparameter $\mathbf{T}$ is estimated based on the EM algorithm described in [8]. In the first iteration, we assume that $\mathbf{T}$ is initialized with random values. Further, given $\mathbf{f}$ and $\mathbf{T}$, $\mathbf{w}$ is estimated as follows

$$\mathbf{w} = \mathbf{L}^{-1}\mathbf{T}^{t}\mathbf{\Sigma}^{-1}\mathbf{f}. \tag{5}$$

$\mathbf{L}^{-1}$ is the covariance estimate of the i-vector and is defined as

$$\mathbf{L} = \left(\mathbf{I} + \sum_{c} \eta_c \mathbf{T}^{(c)\prime}\mathbf{\Sigma}_c^{-1}\mathbf{T}^{(c)}\right), \tag{6}$$

where $\mathbf{T}^{(c)}$ is the c^{th} DxR sub-matrix of T such that

$$\mathbf{T} = [\mathbf{T}^{(1)t}\mathbf{T}^{(2)t}...\mathbf{T}^{(C)t}]^{t}. \tag{7}$$

The E-step in the EM algorithm for estimating $\mathbf{T}$ refers to obtaining the current estimate of $\mathbf{w}_i$ for every utterance $\mathbf{O}_i$ in the development dataset. The M-step involves re-estimating $\mathbf{T}$ as follows

$$\mathbf{T}^{(c)} = \mathbf{C}^{(c)}(\mathbf{A}^{(c)})^{-1}, \tag{8}$$

where $\mathbf{C}$ and $\mathbf{A}$ are defined as follows

$$\mathbf{C} = \left(\sum_{i} \mathbf{f}_i \mathbf{w}_i^{t}\right) \tag{9}$$

and

$$\mathbf{A}^{(c)} = \left(\sum_{i} \eta_{c,i}\left(\mathbf{L}^{-1} + \mathbf{w}_i\mathbf{w}_i^{t}\right)\right)^{-1}. \tag{10}$$

$\eta_{c,i}$ refers to η_c for i^{th} utterance.

The i-vectors contain both speaker and channel information. To get rid of the channel information applying a combination of LDA (Linear Discriminant Analysis) and WCCN (Within Class Covariance Normalization) is studied in [2]. In current state-of-the-art systems, PLDA (Probabilistic LDA) is applied on top of this.

To perform speaker verification, two i-vectors are compared - the i-vector corresponding to the target speaker and the test i-vector. In state-of-the-art speaker verification systems, two i-vectors are compared with respect to the PLDA models estimated from a development dataset and a log-likelihood ratio is obtained as a result [6, 7, 9–11].

A. Differences with the default Kaldi implementation

The implementation of the i-vector system available in the current Kaldi releases can be seen a combination of the standard i-vector implementation mentioned above and the Subspace Gaussian Mixture Model (SGMM) architecture used for Automatic Speech Recognition (ASR). The i-vector extractor estimation algorithm is different from Equations 5 and 8. There are two important differences with the standard implementation.

First, the bias term $\mathbf{m}$ in Equation 1 is subsumed in the matrix $\mathbf{T}$. That is, the first column of $\mathbf{T}$ is $\mathbf{m}$. To remove this offset from, the mean of i-vectors is calculated over a list of i-vectors.

Second, the EM algorithm to estimate $\mathbf{T}$ is based on the SGMM system. In particular, the parameters $\mathbf{\Sigma}_c$ in Equation 6 are updated. The update equations are given in Section 5.7 of [12]. Updating $\mathbf{\Sigma}_c$ does not affect the UBM covariances, however. The posteriors for the sufficient statistics are still computed with respect to the original UBM parameters.

B. Channel Compensation

The i-vectors are low dimensional representations of the audio recordings and still contain information about the channel and the session apart from information about the speaker and the content. To remove these undesirable effects, discriminative classifiers are trained on a development data set. Two levels of classifiers are used. First, the LDA transform is applied and the dimension of the i-vector is reduced. This is followed by PLDA modelling on the dimension-reduced, length-normalized i-vector. Length normalization is discussed in Section II-C. The LDA and PLDA training and scoring algorithms are already available in Kaldi.

C. Length Normalization

In [9], it is observed that length normalized i-vectors after whitening are useful in Gaussianizing the i-vector distribution. This is a simpler form of Radial Gaussianization [13]. This is extremely important when dealing with non-Gaussian nature of the i-vectors, if present [14]. Thus, i-vectors obtained are length normalized after applying the LDA transform. If $\mathbf{w}$ is the i-vector, the length normalized i-vector $\tilde{\mathbf{w}}$ is given by

$$\tilde{\mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||}. \tag{11}$$

III. Scoring

To evaluate the truth of the claims attached to a test utterance, their similarity is measured with respect to the target models. The measure of similarity (or distortion) depends on the modelling technique used. In probabilistic modelling techniques, likelihood scores are evaluated. In deterministic modelling techniques, such as LDA, distance measures are employed. In this section, scoring methods involved in the UBM-GMM framework and the TVS framework are described.

Generally, evaluating a claim consists of validating a hypothesis. Let H_0 be the same speaker hypothesis; that is, both utterances (train and test) belong to the same speaker. This is considered the null hypothesis. This is opposed to the hypothesis H_1 in which the utterances belong to different speakers. Scoring a claim consists of evaluating both hypotheses. To score a claim is to evaluate both hypotheses.

A. Scoring in i-vector framework

Since, every utterance is converted to an i-vector by the system, scoring a claim in a speaker verification system is done using similarity measures between the claimed speaker's

i-vector and the i-vector corresponding to the test utterance. The i-vectors extracted from the supervector contain speaker and channel information. The channel information is removed by discriminative training techniques such as LDA, WCCN and PLDA. Channel compensation using LDA and WCCN are only linear transformations. Moreover, they are deterministic transformations. This is different from PLDA where the transformation is probabilistic in nature. Thus, the scoring mechanism differs in the two cases.

Given two i-vectors to be compared, $\mathbf{w}_1$ and $\mathbf{w}_2$, a cosine similarity measure is used to determine the similarity (or distortion) between them. The cosine similarity measure is computed as follows

$$d_{cos}(\mathbf{w}_1, \mathbf{w}_2) = \frac{\mathbf{w}_1^t \mathbf{w}_2}{||\mathbf{w}_1|| ||\mathbf{w}_2||}, \tag{12}$$

where $||.||$ signifies the norm of the vector. If LDA based channel compensation is applied on the i-vectors, the similarity measure becomes

$$d_{cos}(\mathbf{w}_1, \mathbf{w}_2) = \frac{\left(\mathbf{A}_{LDA}^t \mathbf{w}_1\right)^t \left(\mathbf{A}_{LDA}^t \mathbf{w}_2\right)}{|| \left(\mathbf{A}_{LDA}^t \mathbf{w}_1\right) || || \left(\mathbf{A}_{LDA}^t \mathbf{w}_2\right) ||}, \tag{13}$$

where $\mathbf{A}_{LDA}$ is the LDA projection matrix.

B. PLDA scoring

Scoring in the PLDA framework is different from the previously mentioned scoring methods. The idea here is to verify if the test utterance shares the identity vector with the claimed speaker or they are from two different models. This is summarized as follows

$$s_{cl} = \log \frac{P(\mathbf{w}_{cl}, \mathbf{w}_{tst}|H_0)}{P(\mathbf{w}_{cl}|H_1) P(\mathbf{w}_{cl}|H_1)}, \tag{14}$$

where $\mathbf{w}_{cl}$ is the i-vector of the claimed target speaker, $\mathbf{w}_{tst}$ is the i-vector corresponding to the speaker in the test utterance $\mathbf{O}_{tst}$, H_0 is the same speaker hypothesis and H_1 is the hypothesis that the train and test i-vectors are from different speakers. The method to compute the log likelihood ratio score is based on [15].

IV. Code organization

The code organization follows that in the Kaldi package [16]. The existing code structure in src/ivector and src/ivectorbin is retained. The existing i-vector class naming scheme is followed. The suffix "Conv" is added to the existing classes to refer to the new implementation. For example, the class IvectorExtractor is called IvectorExtractorConv. The entire set of classes are implemented in ivector/conv-ivector-extractor.cc and ivector/conv-ivector-extractor.h The member variables and functions in the classes are similar to those already available in the existing implementation. For some data structures, however, unused variables and functions are removed. The implementation heavily relies on the Kaldi matrix libraries.

A standard Kaldi recipe is available in the scripts folder. This follows the NIST SRE 08 recipe already available so that the results can be easily reproduced and compared. The binaries in src/ivectorbin folder are used by this script. Binaries related to i-vector extraction and EM algorithm to estimate T-matrix can be easily found by appending "-conv" suffix to the default binaries. However, the existing and the new implementations are not compatible with respect to the T-matrix and other relevant parameters.

V. Experiments

The existing Kaldi recipe was modified to test the new implementation with the existing setup. NIST SRE 08 dataset was used as the evaluation set [17]. The evaluation set consists of 8 conditions. The conditions are explained in Table I. The systems are evaluated and compared using Equal Error Rates (EER).

To train the UBM, T-matrix, LDA and PLDA models the following data sets were used: Fisher English Parts I and II, NIST SRE 2004, 2005 and 2006, and Switchboard Cellular Parts I and II. The UBM has 2048 components. The i-vector is configured to have 500 dimensions. The first 150 eigenvectors corresponding to the highest eigenvalues are retained after LDA. Then, length normalization is applied followed by scoring with PLDA. System development and evaluation are gender-dependent. The Kaldi-MFCC features are used by both systems. The features are computed on 20 ms window with 10 ms frame shift with bandpass filtering from 20 Hz to 3700 Hz. The MFCCs have 20 dimensions including the energy component. The energy based speech/non-speech detector is used for voice activity detection.

The results for the male speakers of the dataset are presented in Table II. The EERs are computed using the "compute-eer" tool available in Kaldi. The "Kaldi baseline" system refers to the implementation available in Kaldi and the "Standard i-vector" refers to the Idiap's implementation of the standard i-vector system. Improvements can be observed for conditions 5 and 6. For conditions 2 and 8 there is no change in EER. For all other conditions minor deterioration is observed.

The systems for the female dataset are compared in Table III. Improvements are obtained in more conditions than that seen with the male dataset. In particular, 5 out of 8 conditions showed improvements. In some cases, the gains obtained (for example in condition 1) are significant.

VI. Summary

The standard i-vector system is implemented for the Kaldi toolkit. This is done by modifying the existing codebase for i-vector based speaker recognition. An equivalent recipe for NIST SRE 2008 is provided. The toolkit is available under the Apache 2.0 license at https://github.com/idiap/kaldi-ivector.

VII. Acknowledgement

This work was supported by EU FP7 project Speaker Identification Integrated Project (SIIP).

TABLE I
NIST SRE 2008 CONDITIONS

Condition	Train	Test	Notes
Cond1	Interview speech	Interview speech	All trials
Cond2	Interview speech	Interview speech	Same microphone
Cond3	Interview speech	Interview speech	Different microphones
Cond4	Interview speech	Telephone speech	-
Cond5	Telephone speech	Interview speech	-
Cond6	Telephone speech	Telephone speech	All trials
Cond7	Telephone speech	Telephone speech	English only
Cond8	Telephone speech	Telephone speech	English from native speakers

TABLE II
RESULTS IN TERMS OF EQUAL ERROR RATE (IN %) ON THE NIST SRE 2008 DATASET FOR MALE SPEAKERS.

System	Cond1	Cond2	Cond3	Cond4	Cond5	Cond6	Cond7	Cond8
Kaldi baseline	9.1	1.2	9.3	7.5	7.3	5.4	2.7	2.2
Standard i-vector	9.3	1.2	9.5	7.7	6.6	4.9	3.0	2.2

TABLE III
RESULTS IN TERMS OF EQUAL ERROR RATE (IN %) ON THE NIST SRE 2008 DATASET FOR FEMALE SPEAKERS.

System	Cond1	Cond2	Cond3	Cond4	Cond5	Cond6	Cond7	Cond8
Kaldi baseline	11.5	1.5	11.8	9.8	9.6	6.6	4.3	4.7
Standard i-vector	8.2	0.9	8.1	9.3	8.9	8.3	5.2	5.3

REFERENCES

[1] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support Vector Machines using GMM Supervectors for Speaker Verification," vol. 13. IEEE Signal Processing Letters, 2006, pp. 308–311.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," vol. 19(4). IEEE Tran. on Audio, Speech and Language Processing, 2011, pp. 788–798.

[3] M. McLaren and D. van Leeuwen, "Source-normalized LDA for robust speaker recognition using i-vectors from multiple speech sources," vol. 20(3). IEEE Trans. on Audio, Speech and Language Processing, March 2012, pp. 755–766.

[4] P. Matejka *et al.*, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification." In Proc. of ICASSP, 2011, pp. 4828–4831.

[5] C. M. Bishop, "Pattern recognition," *Machine Learning*, 2006.

[6] O. Glembek *et al.*, "Simplification and optimization of i-vector extraction." In Proc. of ICASSP, 2011, pp. 4516–4519.

[7] S. R. Madikeri, "A hybrid factor analysis and probabilistic pca-based system for dictionary learning and encoding for robust speaker recognition." in *Odyssey*, 2012, pp. 14–20.

[8] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," vol. 13, no. 3. IEEE Trans. on Speech and Audio Processing, May 2005, pp. 335–354.

[9] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." In Proc. of Interspeech, Florence, Italy, August 2011, pp. 249–252.

[10] S. R. Madikeri, "A fast and scalable hybrid fa/ppca-based framework for speaker recognition," *Digital Signal Processing*, vol. 32, pp. 137–145, 2014.

[11] D. Garcia-Romero, X. Zhou, D. Zotkin, B. Srinivasan, Y. Luo, S. Ganapathy, S. Thomas, S. Nemala, G. S. Sivaram, M. Mirbagheri *et al.*, "The umd-jhu 2011 speaker recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4229–4232.

[12] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow *et al.*, "The subspace gaussian mixture modela structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[13] S. Lyu and E. P. SimonCelli, "Nonlinear extraction of independent components of natural images using radial gaussianization," vol. 21, no. 6. Neural Computation, June 2009.

[14] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." Keynote Presentation, Odyssey 2010 - The Speaker and Language Recognition Workshop, 2010.

[15] S. Ioffe, "Probabilistic linear discriminant analysis," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 531–542.

[16] Daniel Povey et al., "The Kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding*, 2011.

[17] "The NIST year 2008 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/sre/2008/index.html.