THE I4U SUBMISSION TO THE 2016 NIST SPEAKER RECOGNITION EVALUATION

Kong Aik Lee, Hanwu Sun, Aleksandr Sizov, Guangsen Wang, Trung Hieu Nguyen, Bin Ma, Haizhou Li
Institute for Infocomm Research, A*STAR, Singapore

Ville Vestman, Md Sahidullah, Ville Hautamäki, Miikka Halonen, Anssi Kanervisto, Tomi Kinnunen
University of Eastern Finland, Finland

Anthony Larcher, Gael Le Lan
LIUM, Universite du Maine, France

Chunlei Zhang, Fahimeh Bahmaninezhad, John H. L. Hansen
CRSS, University of Texas at Dallas, USA

Andreas Nautsch, Sergey Isadskiy, Christian Rathgeb, Christoph Busch
Hochschule Darmstadt, Germany

Themos Stafylakis, Georgios Tzimiropoulos
University of Nottingham, UK

G. Liu, Qi Qian, Zhibin Wang, Q. Zhao, T. Wang, Hao Li, Jian Xue, Shenghuo Zhu, Rong Jin, T. Zhao
Alibaba Inc., USA

Mickael Rouvier, Pierre-Michel Bousquet, Moez Ajili, Waad Ben Kheder, Driss Matrouf, Jean-Francois Bonastre
LIA, University of Avignon, France

Wei Rao, Zhi Hao Lim, Chenglin Xu, Haihua Xu, Xiong Xiao, Eng Siong Chng
Nanyang Technological University, Singapore

Federico Alegre, Benoit Fauve
ValidSoft, UK

Jianbo Ma, Kaavya Sriskandaraja, Vidhyasaharan Sethu, Eliathamby Ambikairajah
University of New South Wales, Australia

M. W. Mak and W. W. Lin
Hong Kong Polytechnic University, Hong Kong

A. K. Sarkar, D. A. L. Thomsen, Z.-H. Tan
Aalborg University, Denmark

Héctor Delgado, Massimiliano Todisco, Nicholas Evans
EURECOM, France

Rahim Saeidi
Aalto University, Finland

Hagai Aronowitz
IBM Research, Israel

ABSTRACT

The I4U's submission to SRE'16 was a result from the collaboration and active exchange of information among researchers across sixteen **I**nstitutes and **U**niversities across **4** continents. The submitted results were based on the fusion of multiple classifiers. A lot of efforts have been devoted to two major challenges, namely, test duration variability and dataset shift from Switchboard and Mixer corpora to the new *Call My Net* dataset.

Index Terms— I4U, SRE'16, Call My Net

1. INTRODUCTION

The consortium (I4U) submission is a joint effort of 16 research Institutes and Universities across 4 continents. The first I4U meeting was conducted via WebEx[1] on May 11, 2016. This was followed by regular bi-weekly and weekly meetings toward the end of SRE'16. An online group was also set up, providing a discussion platform across various issues surrounding NIST SRE'16. In particular, test segment variability, domain adaptation for language and channel shift, uncertainty propagation, score normalization, session compensation, and various issues concerning score calibration (quality measure, supervised versus unsupervised) have been actively discussed. Solutions were put in place as part of the I4U submission. Along the way, we are fortunate to have a group of dedicated researchers joining the I4U efforts toward NIST SRE'16.

Different from previous SREs, the test segments used in SRE'16 were expected to have varying duration ranging from 10 to 60 seconds with uniform distribution. The duration variability was encountered for with the use uncertainty propagation [1] and variance compensated length-norm [2] for i-vector PLDA system. At the score level, we found score normalization (s-norm, t-norm) and the use of quality measure as part of score calibration were useful. The second challenge is dataset shift from Switchboard (SWB) and Mixer datasets used in the previous SREs to the new *Call My Net* dataset used in SRE'16. Two major challenges are the languages mismatch and the changes in data collection infrastructure (e.g., telephone network, front-end devices). In particular, the SWB and Mixer corpora were mainly English utterances collected in North America, while the *Call My Net* corpus consists of Cebuano, Tagalog, Mandarin, and Cantonese utterances collected in Asia Pacific. To this end, we found the *Inter Dataset Variability Compensation* (IDVC) [3] was extremely effective. Most sub-systems developed by I4U were equipped with the duration and dataset compensation techniques mentioned above.

[1]Thanks Nick Evans and his team for facilitating the conference call.

2. PRIMARY AND CONTRASTIVE SUBMISSIONS

I4U submitted three systems to SRE'16 *Fixed Training Condition*, one primary and two contrastive, with different classifier selection and fusion strategies. The primary system is a linear fusion of 17 sub-systems selected from a pool of 33 sub-systems contributed by I4U members. The fusion parameters were trained using logistic regression fusion implemented in the BOSARIS toolkit [4]. The first contrastive system is a fusion of 4 sub-systems with a simple unsupervised score calibration using unlabeled dataset. The second contrastive system is a fusion of all 33 subsystems. A slightly different fusion strategy was used, as detailed in Section 5. The 17 sub-systems used in I4U primary submission are listed in Table 1.

It is worth mentioning that almost all the sub-systems listed in Table 1 are based on i-vector [5], which represents the mainstream technique in text-independent speaker recognition. In addition, a rich set of acoustic feature extraction front-ends were used in our submission, which include MFCC, Bottleneck Feature, Tandem feature, and ICMS (Section 4.14). At the i-vector extraction stage, we have sub-systems that use either UBM or DNN posteriors [2, 6]. In addition to i-vector, we also experimented with GMM-UBM and GMM-SVM which we included in the contrastive system. Detailed description is provided in the system description presented in Section 4.

Table 1. *Subsystems used for I4U primary submission*

	Component classifiers	Site
1.	PNCC IV-PLDA + GMM-UBM	AAU
2.	Alibaba4:Symmetric SVM	Alibaba
3.	CRSS1:UBM IV LDA PLDA	CRSS
4.	CRSS2:UBM IV LDA PLDA	CRSS
5.	CRSS3:UBM IV SVDA PLDA	CRSS
6.	ICMC IV PLDA	EURECOM
7.	hda4	HDA
8.	I2R MFCC IV-PLDA	I^2R
9.	I2R Tandem DNN IV-PLDA	I^2R
10.	LIUM MFCC DNN I-vector	LIUM
11.	NTU1 Tandem IV-PLDA	NTU
12.	polyu3 IV-PLDA	HK Poly U
13.	UEF1 MFCC i-vector-PLDA	UEF
14.	UNSW MFCC i-vector/PLDA	UNSW_EET
15.	VSO2 PLP i-vector/PLDA	ValidSoft
16.	UoNottingham DNN i-vector/PLDA	Nottingham
17.	LIA MFCC GMM i-vector/PLDA	LIA

3. TRAIN AND DEVELOPMENT DATASETS

Table 2 lists datasets that we used for the parameter training, optimization of the component classifiers and the fusion. All of these datasets were provided by NIST and LDC for Fixed Training Condition. In particular, the Call My Net (LDC2016E46_SRE16_Call_My_Net_Training_Data) and SRE'10 were used as the development-test set.

Table 2. Corpora

Corpus	Task
LDC2016E46_SRE16_Cal_My_Net	Development Test Set 01
LDC2016E46 (Unlabeled subset)	Score normalization, Score calibration
SRE10 CC5	Development Test Set 02
SRE04, 05, 06, 08 Switchboard-2 Phase II Switchboard-2 Phase III Switchboard Cellular Part 1 Switchboard Cellular Part 2	UBM, T matrix, PLDA
Switchboard-1 Release 2 Fisher 1, Fisher 2	DNN

4. COMPONENT CLASSIFIERS

Within the I4U consortium, participating sites contribute to the SRE'16 submission in one form or another. Listed below are the descriptions of the component classifiers used for I4U submission.

4.1. Institute for Infocomm Research (I^2R)

I^2R contributed two sub-systems to I4U submission. Presented below is a description of various components in the pipeline from feature extraction, i-vector extraction, preprocessing to PLDA scoring used in the two subsystems. More details can be found in [7].

MFCC I-VECTOR – The MFCC feature vectors consist of 20 static coefficients together with their first and second delta coefficients. A simple energy-based VAD was used based on the C0 component of the MFCC feature. The algorithm is based on thresholding the log-mel energy and taking the consensus of threshold decisions within a window of 11 frames centred on the current frame. The 60-dimensional MFCC feature vectors were then used to train the UBM and T matrix. A randomly selected set of 16,000 utterances were used to train a UBM with 2048 Gaussians with diagonal covariance matrices. Based on the diagonal-covariance UBM, a full-covariance UBM with 2048 components was trained on a randomly selected set of 32,000 utterances. All utterances were used to train the i-vector extractor with the T matrix of a matrix rank of 600. The entire chain from MFCC feature to i-vector extraction was implemented using the KALDI toolkit [8].

DNN I-VECTOR – A deep neural network containing a bottleneck layer was trained to generate 80-dimensional bottleneck feature vectors using the KALDI toolkit [8]. They were then used together with another set of 57-dimensional MFCC features [9]. The concatenated features (137 dimensions) are referred to as tandem features. The bottleneck DNNs were trained using the fisher and switchboard landline data, which all come with transcriptions. The DNN was trained to predict 8724 senones. The input features are 43-dim consisting of 40-dim filter bank features and 3-dim pitch features. Delta and delta-delta coefficients were computed using the 43-dimensional raw features yielding a 129-dimensional feature vector. The features were then expanded to a 21-frame window to take account of the speech context leading to an input layer of 2709 units. The DNN contains 7 hidden layers of 1024 hidden units with RELU activation function except the 3rd hidden layer, which has 80 units with linear activation function serving as the bottleneck layer. To extract the posterior features, another DNN was trained with the same topology as the bottleneck DNNs but trained as a p-norm [10] DNN using only the switchboard landline data (with 8797 senones). To save hard disk and memory usage, for each frame, we only output the top 88 posteriors. The DNN posteriors were used in place of GMM posteriors in the i-vector extractor. We also performed senone tying [4] to reduce the number of tied Gaussian components to 2395. We set the rank of T matrix to 400 for DNN i-vector.

I-Vector PLDA – Pre-processing of i-vectors consists of the following steps: IDVC [3], PCA, whitening, and projection-to-unit-sphere. For IDVC, we used 8 gender-language subsets of the unlabelled data and 8 gender-corpus subsets of SRE'04, 05, 06 and 08. After IDVC we projected the data using PCA into lower dimensional space to physically remove those 16 dimensions that we cancelled by IDVC. We used simplified PLDA for the scoring. To this end, we trained PLDA model on SRE data only. We did separate training for both i-vector types, then applied s-norm (using the unlabelled dataset).

CPU time and memory – Major part of the computation was dedicated to the bottleneck feature extraction, DNN posterior inference, MFCC and i-vector extraction. Table 3 shows the CPU time (single threaded) and the memeory consumption per trials. In our implementation, We parallelized the computation on 64-bit machines with Intel Xeon E5-2685 v3 @ 2.30 GHz CPU with 56 CPUs and 256G of RAM. DNN computation was carried out on a Tesla K20m GPU with 4742MB memory. I-vector scoring on a single core of Intel i7-4790 @ 3.60 GHz CPU.

Table 3. *I^2R sub-systems – CPU time (single threaded) and memory used to process a single trial (one enrollment and one test segment) from acoustic feature extraction, to i-vector and PLDA scoring*

Processes/submission	CPU time (sec.)	Memory (MB)
DNN BNF Extraction	5.3110	774
DNN Posterior Estimation	183.7091	1606
MFCC Extraction	3.2949	3
I-vector extraction	10.4147	4016
PLDA scoring	0.0008	50

4.2. University of Eastern Finland (UEF)

UEF provided three subsystems (UEF1, UEF2, UEF3) for the I4U submissions. All three systems are based on the same i-vectors but they differ in i-vector processing and scoring. From these systems, only the first one (UEF1) is included in the primary submission, while all of them are included in the contrastive system of all subsystems.

Feature vectors are 60-dimensional consisting of 20 base MFCC coefficients (including energy coefficient) and their first and second time derivatives. The features were computed from 20ms long Hamming windowed frames with 10ms overlap. Speech activity detection was performed by using energy thresholds obtained from bi-Gaussian modeling of log energies. A little tweak was made for the UEF1-system: For the SRE16 data (enrollment, test, unlabeled), the speech activity thresholds were set based on the maximum of the frame energies instead of bi-Gaussian modeling.

Data from SRE04-SRE06, Switchboard, and Fisher corpora was used to train a 1024 component UBM with diagonal covariance matrices and a total variability matrix for extracting 600 dimensional i-vectors. In the case where three enrollment segments are given per speaker, extracted i-vectors were averaged to have only one i-vector per speaker.

UEF1: Inter dataset variability compensation (IDVC) [3] was performed to all i-vectors using mean parameters only (subspace dimension = 3). This reduced i-vector dimensionality to 597. Four different datasets were used in IDVC: two from Switchboard corpora and two others consisting of unlabeled minor and major data. As the last step of i-vector processing, they were centered, length-normalized, and whitened. Then, a simplified PLDA model with latent variable dimensionality of 200 was trained using SRE04-SRE08 data. After PLDA scoring, symmetric normalization (s-norm) [11] was applied to the obtained scores. For the normalization, 1000 impostor i-vectors were randomly selected from the background data (SRE04-08 + Switchboard).

UEF2: The dimensionality of i-vectors was first reduced to 200 by using linear discriminant analysis. The data for LDA training was taken from the SRE04-08 and Switchboard corpora. Then, the i-vectors were whitened with the unlabeled data and length normalization was performed. Trials were conducted using cosine similarity scoring.

UEF3: The i-vectors were processed in the same way as in UEF2. For this system, a support vector machine with linear kernel was trained using data from SRE04-08 and Switchboard corpora.

4.3. LIUM

LIUM contributed three sub-systems to I4U submission. Presented below is a detailed description of various components in the pipeline from feature extraction, i-vector extraction, pre-processing to PLDA scoring used in the three subsystems. All LIUM sub-systems have been developed using SIDEKIT [12]; documentation and scripts are available at http://lium.univ-lemans.fr/sidekit

MFCC GMM I-VECTOR

Static coefficients consist of 19 dimensional MFCC plus the log-energy. After adding their first and second derivative, RASTA filtering is applied, selection of frames based on the energy and CMVN are applied. The i-vector extractor is made of a 2048 GMM with diagonal covariance and a Total Variability matrix of rank 500. 29,301 sessions from SWB, NIST-SRE 04, 05, 06 and 08 are used to train the TV matrix and a subset of 647 sessions are used for the UBM (329 female /318 male sessions).

MFCC DNN I-VECTOR

A 5-layer Neural Network (1200-1200-80-1200-1200) using sigmoid activations is trained with SIDEKIT linked to Theano [13]. It is used to compute the frame alignments on the 2,304 senones of the soft-max layer of the network. Input of the network are 15 acoustic frames (7 + 1 + 7). A fake 2,304 distribution GMM-UBM is obtained using 647 sessions [6] and a classic i-vector extractor of rank 400 is build on top.

TANDEM GMM I-VECTOR

previously described MFCC are concatenated with 80 dimensional that extracted using the same neural network. CMVN is applied per utterance on the tandem features and VAD is shared with other systems. The i-vector extractor consists of a 1024 distribution GMM-UBM with diagonal covariance and a TV matrix of rank 400.

Back-end

All three LIUM subsystems share the same backend. I-vectors are whitened and length-normalized with parameters estimated on 34,218 telephone sessions from SWB, SRE04, 05, 06 and 08.A scaling factor of 0.5 is used during training and enrollment phase. The mean of the PLDA is replaced

by the mean of the Call-My-Net development data. Multiple utterances of a same speaker are averaged. Uncertainty propagation is used only with test utterance uncertainty [1].

Timing information and more details can be found in LIUM's system description.

4.4. CRSS

CRSS1: UBM i-Vector LDA PLDA. This system is mainly modified version of Kaldi (sre10/v1). 60 dimensional feature vectors for each frame is adopted here including 20 dimensional MFCC features appended with $\Delta + \Delta\Delta$. Unvoiced parts of the utterances are removed with energy based voice activity detection (VAD). For training 2048-mixture UBM and total variability (TV) matrix, SRE2004, 2005, 2006, 2008, telephone data of SRE 2010, Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2 (SWB) and Fisher English are used. Next, 600 dimensional i-Vectors are extracted and their dimensions are reduced to 580 with LDA. For training LDA/PLDA, only SRE 04-08 are used; in addition, speakers who have less than 4 utterances is filtered out. Also, unsupervised speaker clustering is performed; 75 speaker clusters for unlabeled minor data and 300 for unlabeled major data are generated. Before PLDA scoring, mean subtraction is also applied.

CRSS2: UBM i-Vector LDA PLDA. An alternative UBM i-Vector system also adopted from Kaldi (sre10/v1). In this system, feature vectors contain 20 MFCCs appended with $(\Delta+\Delta\Delta)$ coefficients. The window length and shift size are 25-ms and 10-ms, respectively. In addition, we did cepstral mean normalization using 3-sec sliding window. Next, Non-speech frames are discarded using energy-based voice activity detection. 2048-mixture full covariance UBM and total variability matrix have been trained using data collected from SRE2004, 2005, 2006, 2008 and Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2. At the back-end, after extracting i-Vectors, the global mean calculated from minor and major unlabeled data is subtracted from all i-Vectors. Next, i-Vectors are length-normalized and their dimension are reduced from 600 to 400 using LDA. Again, i-Vectors are length-normalized. Finally, trial-based mean subtraction is applied (the participant i-Vectors in a trial are averaged and the value is subtracted from both i-Vectors) and scores are calculated using PLDA. The front end is trained with SWB and SRE04-08; however, the back-end only uses SRE04-08. For back-end mostly MSR [14] toolkit has been adopted.

CRSS3: UBM i-Vector SVDA PLDA [15]. The i-Vectors are the same with CRSS2 system, the only difference between CRSS3 and CRSS2 is the discriminant analysis method used for dimension reduction. In this system, LDA is replaced with discriminant analysis via support vectors (SVDA). SVDA uses only distinct support vectors to calculate the between and within class covariance matrices. For training SVM classifier in SVDA framework, 1-vs-rest strategy has been chosen; since, we wanted to use unlabeled minor and major data without needing their labels. Therefore, when one class is classifying against the rest, the minor and major unlabeled data are added to the rest class. SVDA reduces the dimension of i-Vectors from 600 to 400. More details on SVDA can be find in [15].

CRSS4: DNN i-Vector LDA PLDA. This is a DNN i-Vector system using Kaldi (sre10/v2) based on the multi-splice time delay DNN (TDNN) [16]. TDNN is trained with only a small portion of Fisher English data (1239 utterances). The feature vectors contain 40 dimensional f-bank features. TDNN has six layers; the hidden layers have an input dimension of 350 and an output dimension 3500. The softmax output layer computes posteriors for 3859 triphone states. More details on the TDNN structure and training procedure are provided in [16]. After TDNN training, 20 MFCCs appended with $(\Delta + \Delta\Delta)$ coefficients (overall 60 features) are employed for training TV matrix. Next, 600-dimensional i-Vectors are extracted. After i-Vector extraction, we apply similar strategies for back-end such as LDA and PLDA, briefly described in CRSS2.

4.5. Hochschule Darmstadt (HDA)

HDA contributed two sub-systems to I4U submission (hda2 & hda4), which are based on four processing schemes and fourteen quality metrics employed in a quality-informed fusion scheme [4] prior to I4U fusion. HDA i-vector/PLDA systems contributed to I4U are based on two LIUM i-vector extractor front-ends [17]: MFCC24 clustered by a 4096-component UBM for extracting 600-dimensional i-vectors, and MFCC40 clustered by a 2048-component UBM for extracting 500-dimensional i-vectors. The MFCC24 front-end extracts 12 MFCCs and the log-energy, whereas the MFCC40 front-end extracts 19 MFCCs and the log-energy. First and second order derivatives are computed using CMVN [17]. All implementations rely on SIDEKIT [12] and BOSARIS [4].

hda2: the conventional i-vector/PLDA comparator utilizes MFCC24 i-vectors. Spherical projection is conducted by LDA projection to 400 dimensions, WCCN and length-normalization. Gaussian PLDA with a 250-dimensional speaker sub-space is employed carrying out comparisons. LDA, WCCN and PLDA are trained on previous SRE data. Neither score normalization, nor pre-calibration are employed.

hda4: a pre-fused quality-informed system based on three domain-adapted processing schemes. After reduction to 400 dimensions by LDA, the processing schemes conduct whitening instead of WCCN (mean shift & rotation). The domain adaptation is trained on all unlabeled SRE'16 development data. Two processing schemes are based on the MFCC24 front-end employing two covariance (2Cov) and PLDA (250-

dimensional sub-space) comparators, respectively. The third processing scheme is based on the MFCC40 front-end, and accounts for uncertainty during the i-vector extraction in terms of employing Uncertain LDA (ULDA) [18] instead of LDA for projecting i-vectors to a 400 dimensional biometric feature space. ULDA comparisons are carried out by 2Cov. The hda4 sub-system is pre-fused and calibrated using BOSARIS' quality-informed calibration scheme [4] and the seven divergence and no-reference quality estimates depicted in [19]. Pre-fusion is carried out on the full SRE'16 labeled development set.

Quality estimates: our sub-systems estimate the "acoustic richness"[2] underlying to an i-vector extraction process. Thereby, the top-1 scoring UBM component is tracked resulting in a component lattice, assuming UBM components cluster acoustic groups of similar properties e.g., phonemes in a generalized point of view. For the purpose of utilizing lattice-based quality estimates in a quality-informed calibration scheme, quality divergence and no-reference quality are estimated and reduced to one or two scalar representations, respectively. Quality estimates are based on 60 MFCCs including log-energy, first, and second order derivatives, extracted from a MFCC24 front-end. For the purpose of deriving quality estimates, lattices are interpreted as graphs, i.e. adjacency matrixes, or as histograms. Adjacency matrixes are examined regarding their spectra [20, 21], i.e. eigenvalues, and the Bhattacharrya similarity of corresponding covariance representations [22]. Histogram information is examined in terms of the Bhattacharyya coefficient, the dot product, and the normalized McNemars Chi-squared test. I-vector posterior covariances are compared by the Jensen-Bregman LogDet divergence [23].

Timing and preliminary performance reporting are depicted in Tab. 4. Front-ends computations were carried out on LIUM servers, comparisons were conducted on Intel(R) Xeon(R) CPU E5-2698 v3 2.30GHz, and pre-calibration and fusion were performed on Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz. Performance reporting is referred to the full labeled development set.

System	Performance	Enrol	Verify
hda2	22.79 / 0.90 / 8.77	0.30	0.30
hda4	16.91 / 0.72 / 0.78	0.57	0.57

Table 4. Preliminary performance (EER / minDCF'16 / actDCF'16), and execution time reporting as a portion of real-time.

4.6. University of Nottingham

University of Nottingham contibuted two subsystems. Both of them were developed on SIDEKIT ([12]) and the i-vectors were generated by Anthony Larcher from LIUM (see the corresponding section for details regarding training sets and CPU execution time). Subsystem 1: UoNottingham GMM-iVector/PLDA with Uncertainty propagation This subsystem uses a UBM-based i-vector representation with MFCC. MFCC features are extracted with the following configuration. After pre-emphasis filtering, 40 Mel-scaled filters are used to extract 19 MFCCs and the log-energy of each frame. First and second derivatives are computed and normalized using per-utterance CMVN. The UBM has 4096 components with diagonal covariance matrices, while the i-vectors are 400-dimensional. The i-vectors are length-normalized with prewhitening statistics (means and covariance) estimated on the Call-My-Net unlabelled corpus. A scaling factor of 0.4 is used during training and enrollment phases, in order to increase the uncertainty of the speaker statistics. Moreover, the mean of the PLDA is set equal to the mean of the Call-My-Net unlabelled corpus. In cases of speaker models with 3 enrollment utterances, i-vector averaging is applied without changing the second order statistics of the speaker model. Finally, uncertainty propagation is used, taking into account only the uncertainty of the test utterances [1]. The results on the development set are: EER = 18.46 % and minCprimary = 0.706. In terms of minCprimary, the most effective idea is the scaling factor, which leads to 0.05 absolute improvement. Uncertaity propagation is doing well in terms of EER but slightly degrates the performance in terms of minCprimary. Finally, the use of Call-My-Net data for prewhitening and PLDA centering is very effective in both EER and minCprimary.

Subsystem 2: UoNottingham DNN-iVector/PLDA This subsystem uses a DNN for Baum-Welch statistics estimation. The DNN has 2304 senones and is trained on Switchboard corpora. The frame posteriors are estimated using the DNN and are combined with MFCC frames (described above) to extract Baum-Welch statistics [6]. Again, the prewhitening of the i-vectors is performed using first and second order statistics estimated on the Call-My-Net unlabelled corpus. The backend is a PLDA model with scaling factor of 0.4. However, we did not apply uncertainty propagation, due to a notable degradation on the Cprimary. The subsystem was included in the I4U primary submission. The results on the development set are: EER = 19.67 % and minCprimary = 0.695. Again, the use of the scaling factor contributed about 0.04 absolute improvement on minCprimary, while the prewhitening and PLDA centering based on Call-My-Net data were both helpful (about 0.03 absolute improvement on minCprimary).

4.7. Alibaba

Alibaba1: DNN-iVector PLDA: Kaldis recipe (sre10/v2) is adopted [8]. iVector dimension is 600. It is reduced to 370 dimension with LDA. All iVectors are subtracted mean of the all the 2472 unlabeled data of SRE2016.

[2]Term coined by Rahim Saeidi during pre-SRE'16 disussions.

Alibaba2 and Alibaba3: GMM-iVector PLDA: Kaldis recipe (sre10/v1) is adopted [8]. For Alibaba5 and Alibaba6, iVector dimension is set as 600 and 800, respectively. It is reduced to 400 dimension with LDA. All iVectors are subtracted mean of the all the 2472 unlabeled data of SRE2016.

Alibaba4: Symmetric SVM: It is shown SVM based backend can boost the speaker recognition performance [24]. Similar framework as followed here with some modification. Using iVectors from DNN-iVector PLDA (Alibaba2) framework as features, two SVM model is built for each trial. It is known that each trial involves an enrollment speaker and test segment. In the first SVM, the mean of all the data samples from the involved enrollment speaker is used positive sample, all the iVectors of the unlabeled data are used as negative samples. In the second SVM, the iVector of the test segment is used as positive sample, all the iVectors of the unlabeled data are used as negative samples. Linear kernel is adopted. It is based on LibSVM [25]. The output probability is converted into log likelihood with logarithm operation.

4.8. University of Avignon (LIA)

LIA system is based on i-vector/PLDA paradigm [5]. The front-end is based on MFCC features. MFCC features consist of 20 dimensional MFCC with their delta and delta-delta. A simple energy-based VAD was used on the C0 component of the MFCC feature. The MFCC are used to train an UBM and T matrix. The size of the UBM is 4096 and the i-vectors are 400-dimensional. For UBM and T matrix training all SRE04-08 and SWB are used. For PLDA training and i-vectors normalization only SRE04-08 is used. The i-vectors are length-normalized with prewhitening statistics (means and covariance) estimated on the train corpus [26, 27]. Inter dataset variability compensation (IDVC) [3] is applied to remove the inter-dataset variability. We used six different datasets in IDVC based on language and gender and the additional subset of development data from the major language provided by NIST (the latter only for mean-subspace removal, as this subset is unlabeled). Then the between- and within-covariance matrix are estimated with PLDA. A post-PLDA normalization procedure is proposed that simultaneously diagonalizes between- and within-class covariance. Finally, a new length-normalization is applied.

4.9. Nanyang Technological University (NTU)

NTU system is based on i-vectors/PLDA framework. Tandem features are used to replace traditional MFCCs for extracting i-vectors.

4.9.1. Front-end

The features of this system are based on 90-dim Tandem features which are formed by MFCCs and bottleneck features (BNFs). For MFCCs, we adopt the standard MFCC extraction process. 19 MFCCs together with energy plus their 1st- and 2nd-derivatives are extracted, which leads to 60-dimensional features. For BNFs, we select switchboard data provided by NIST to train the stacked bottleneck neural network [28] (BNFs extractor), then extract the 30-dimensional BNFs from the given speech file. Finally, we concatenate 60-dim MFCCs and 30-dim BNFs at frame level to obtain 90-dim tandem features. The tandem features are processed by cepstral mean normalization with a window size of 3 seconds. An energy based voice activity detection (VAD) method is used to remove the silence frames.

4.9.2. I-Vector Extraction

We select 71,917 utterances from SRE16 switchboard training data, NIST SRE04 SRE08 to train UBM with 2048 Gaussians. The same dataset is used to train a total variability matrix with 400 total factors.

4.9.3. Channel Compensation and Scoring

Because data in SRE16 is obtained from CallMyNet that is quite different from the past NIST SRE data, Inter dataset variability compensation (IDVC) [5] was applied to remove the inter-dataset variability. After IDVC, PCA is applied to transform 400-dim i-vectors into 350-dim i-vectors. Length normalization is performed on the PCA projected i-vectors. Then, we perform LDA on the resulting i-vectors to reduce the dimension to 200 before training the PLDA models with 150 latent variables. The evaluation scores are obtained by PLDA scoring method.

4.9.4. Spectral Clustering

2,472 utterances from CallMyNet are provided by SRE16 to improve the system. But gender, speaker, language labels are not provided. This will cause problems to adopt LDA and PLDA. Therefore, we adopt spectral clustering to obtain the estimated speaker label for these unlabelled utterances. Spectral Clustering algorithms are a class of segmentation techniques that are based on the eigenvectors of the affinity matrix. The main intuition behind this approach is that the affinity matrix of the data contains information of their clusters in the data set.

After obtaining the i-vectors of the unlabelled data, LDA projection which is estimated from dataset with true speaker label is performed to reduce the dimensions from 400 to 200. Then, the projected i-vectors are used to create the laplacian based on [29] and orthogonalised according to [30]. Following the main algorithm described in [30], spectral clustering is applied by setting the numbers of clusters set to 240. This number of clusters was estimated by assuming that each speakers have about 10 utterance each. The radial basis function (RBF) [31] was used as a similarity measure. The obtained cluster (speakers) labels are used to train PLDA model.

4.9.5. Score normalization

We select 200 unlabelled major language utterances for test normalization.

4.9.6. Computation Time and Performance

NTU system can be divided into two parts: front-end processing and back-end processing. The former includes tandem feature and i-vector extraction, which is based on Kaldi toolkit. The latter includes channel compensation, scoring, and score normalization, which is performed by MATLAB. The experiments on front-end and back-end processing are conducted on an Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz and an Intel(R) Xeon(R) CPU E5-2687W @ 3.10GHz, respectively. Because NTU system is based on i-vector framework, the steps and CPU time for processing enrolment and test segment are similar. The CPU time of front-end processing per utterance is around 5.26 s. and that of back-end processing per trial is around 0.007 s. The EER and minimum Cprimary of NTU system on SRE16 development set are 17.33 and 0.734, respectively.

4.10. ValidSoft (VSO)

The two subsystems denoted VSO1 and VSO2 that are reported by ValidSoft are based on an UBM i-vector representation with 50-dimentional PLP-based feature vector (19 static plus selection of delta and delta-delta coefficients). The UBM consist in 512 Gaussians trained on NIST SRE 04 data.

The T matrix is estimated using NIST SRE 04-05-06 and switchboard data in order to extract i-vectors of dimension 600. Three-iteration Eigen Factor Radial (EFR) normalization are applied to i-vectors prior to dimensionality reduction down to 400 via LDA, followed by PLDA scoring. The global mean calculated from minor and major unlabelled data is subtracted from all i-vectors prior to LDA and PLDA processes. The audio segments used as background data for the i-vector back-end (NIST 04-05-06-08) are processed through the ITU-T G.729 ANNEX-A codec usually used in VoIP.

For VSO1, NIST 05-06 is not used in the backend processing, while for VSO2 the global mean of the PLDA data is replaced by the global mean calculated from minor and major unlabelled data before subtraction.

4.11. University of New South Wales (UNSW)

The sub-system is an i-vector G-PLDA system with a denoising autoencoder applied to the i-vectors. The front-end of the sub-system comprises of 13 dimensional MFCC features along with their first and second derivatives estimated in conjunction with a vector quantization model based voice activity detector [32] prior to feature warping [33]. A gender-independent universal background model (UBM) of 2048 Gaussian mixtures was created using 4802 utterances from a background set comprising of NIST SRE04, 05, 06, 08, Switchboard II Part 1, 2, 3 and Switchboard Cellular Part 1 and 2 databases. In selecting the data for training the UBM, one utterance was chosen from each speakers available data to retain speaker diversity while reducing the overall amount of data [34]. A T matrix of rank 600 was estimated [5] using 68706 utterances from the 4802 speakers. i-vectors were computed for each of the background, NIST SRE 2016 development and evaluation set using the estimated T-matrix. LDA was then applied to further reduce the dimension to 400 following which the i-vectors were radial Gaussianised and length normalised [35]. A denoising autoencoder [36] of 1000 neurons was trained on i-vectors from short duration utterances (10, 20, 30, 40, 50 seconds) obtained by truncating full length utterances to map the i-vector space of noisy short duration utterances to the i-vector space of clean full length utterances. This denoising autoencoder was used to transform i-vectors from the background, NIST SRE 2016 development, and evaluation sets. A G-PLDA was then trained on top i-vectors from the background set and unlabelled data from development set. Weighted likelihood based domain adaptation [37] was applied during G-PLDA training by assuming each utterances of unlabelled data came from different speakers. Finally, in the evaluation stage, the mean of G-PLDA was replaced by a mean that was calculated from the unlabelled data.

4.12. Hong Kong Polytechnic University (HK Poly U)

4.12.1. Acoustic Features

Speech regions in the speech files were extracted by using a two-channel VAD [38]. For each speech frame, 19 MFCCs together with energy plus their 1st and 2nd derivatives were computed, followed by cepstral mean normalization and feature warping [33] with a window size of 3 seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

4.12.2. I-vector Extraction and PLDA Model Training

The i-vector/PLDA system is based on a gender-independent UBM with 512 mixtures and a gender-independent total variability matrix with 300 total factors. Unlabelled utterances from CallMyNet development data were used for training the UBM and total variability (TV) matrix. The TV matrix and UBM were used for extracting i-vectors from the speech files (both gender) in Switchboard-2 Phase I to Phase III, Switchboard Cellular Part 1 and Part II, and NIST 2004–2010 SREs. Utterances with bad recordings (e.g., without speech or contain tone only) as detected by the VAD and utterances with speech frames less than 10s were excluded. Speaker-to-utterance mappings were determined from the key files of these corpora, with the identical speaker IDs across multiple speech corpora considered to be the same speakers.

Speakers with less than 4 speech segments were excluded. This amounts to 66,505 speech segments (i-vectors) spoken by 4,959 speakers.

Following [39], within-class covariance normalization (WCCN) [40] and i-vector length normalization [35] were applied to the 300-dimensional i-vectors. Then, linear discriminant analysis (LDA) [41] and WCCN were applied to reduce the dimension to 200 before training an unadapted gender-independent PLDA model with 200 latent variables.

4.12.3. Domain Adapation

To make the PLDA model amenable to CallMyNet data, the following domain adaptation procedures were applied. First, pairwise PLDA scores of unlabelled utterances in the CallMyNet development set were computed. Then, spectral clustering [42] was applied to the resulting pairwise scoring matrix to cluster the i-vectors in CallMyNet into 300 clusters.[3] The i-vectors of these 300 hypothesized speakers were than added to the pool of training i-vectors to retrain the PLDA model.

The following whitening step was also applied to make the i-vectors of target-speakers and test utterances better reflecting the acoustic characteristics of CallMyNet data. Specifically, the mean of the unlabelled CallMyNet i-vectors was subtracted from each of the target-speakers' and test i-vectors before applying i-vector pre-processing (WCCN whitening, length-normalization, and LDA-WCCN projection).

4.12.4. PLDA Scoring and Score Normalization

According to the evaluation protocol, for each evaluation trial, a test utterance was tested against the target-speaker's i-vectors representing the Model ID of that trial, which produces one or multiple PLDA scores. The average of these scores is considered as the trial score.

To reduce the actual DCF, the PLDA scores were normalized by T-norm and S-norm [11]. 400 utterances (i-vectors) from the unlabelled segments in CallMyNet were used for T-norm, and another 400 were used as impostor utterances for Z-norm.

4.12.5. Performance and Computation Time

Table 5 shows the performance (in terms of EER, minDCF and actual DCF) of three systems in the development set of SRE16. In the table, Sys A, Sys B, and Sys C represent the system without score normalization, with T-norm and with S-norm, respectively.

Table 6 shows the CPU time and memory requirements for computing the score of one verification trial. Tasks 1–3 were implemented in C and Tasks 4–6 were implemented in Matlab. The memory consumption in the Matlab tasks includes the memory of the Matlab shell without the GUI.

[3] While the number of speakers in the development data of CallMyNet is much smaller than this value, we found that performance is better if we set this value higher than the actual number of speakers.

4.13. Aalborg University (AAU)

PNCC IV-PLDA, GMM-UBM: This system is based on the Power-normalized Cepstral coefficients (PNCC) [43] feature. The frontend processing applies spectral subtraction [44] before the PNCC features (13 Static with Δ+$\Delta\Delta$) are extracted, VAD labels from the VQVAD [32] algorithm are used for both frame dropping and the enhancement. Average scores of GMM-UBM [45] with t-norm (having 512 mixtures, trained using data from SRE 04, 05, 06, Switchboard and unlabeled SRE 16 development) and i-vector [5] PLDA [46] based systems are utilized. Total variability space is built using data from SRE 04, 05, 06, 08 and Switchboard corpora. ALIZE [47] and BOB [48] toolkits are used to implement the respective systems. I-vectors (400 dimensions) are conditioned by 1-iteration of eigen factor radial (EFR) [26] algorithm before PLDA. Un-label (major, minor) development data is used for implementation of t-norm, EFR and mean (μ) parameter in PLDA. In the i-vector system, PLDA is trained using i-vector for various duration of speech segments (from SRE 04-08 database) to reduce the effect of data mismatch duration between the training and testing phases of speaker verification. Target models are represented by number of i-vectors which are derived by segmentation of their respective training utterances independently (number of segment depends on the length of respective speech utterance). The inclusion of data segmentation yields a performance increase on the dev system. In test, average score is computed over target specific i-vectors for a given i-vector of the particular test utterance. In the case of the GMM-UBM system, target models are derived from the GMM-UBM using their respective training data with three iterations of MAP adaptation. Only Gaussian mean vectors of the GMM-UBM are adapted during MAP. The experiments run on a Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GH, with a CPU time of 5.36s per utterance for frontend processing and feature extraction. The CPU time for the training and testing with the GMM-UBM is 4.55s, while the I-vector-PLDA system takes 20.78s.

4.14. EURECOM

EURECOM ICMC-iVector, PLDA: a system which uses infinite impulse response - constant Q, Mel-scaled cepstral coefficients (ICMC) [49]. ICMC feature extraction draws upon the constant Q transform (CQT) which has a variable spectro-temporal resolution with a greater frequency resolution at low frequencies and a greater time resolution at high frequencies. ICMC is applied with a frame shift of 10ms, a Q factor of 96 and RASTA filtering to extract 19 static + log-energy, delta and delta-delta features giving vectors of dimension 60. Non-speech frames are then removed with an energy-based VAD

Table 5. Performance of three ivector-PLDA systems in the development set of SRE16. *Sys A*: No score normalization; *Sys B*: T-norm; *Sys C*: S-norm. *mDCF*: Minimum DCF; *aDCF*: Actual DCF

Sys	Mandarin (CMN)						Cebuano (CEB)						CMN+CEB		
	Male			Female			Male			Female			Male+Female		
	EER(%)	mDCF	aDCF	EER(%)	mDCF	aDCF	EER(%)	mDCF	aDCF	EER(%)	mDCF	aDCF	EER(%)	mDCF	aDCF
A	5.36	0.458	0.714	17.34	0.819	4.905	22.68	0.866	1.806	23.29	0.956	4.734	17.49	0.775	3.040
B	7.39	0.442	0.674	19.01	0.797	0.842	25.30	0.811	0.955	24.15	0.935	0.975	20.89	0.746	0.862
C	6.03	0.461	0.748	17.00	0.781	0.875	23.36	0.823	0.975	23.61	0.945	0.980	18.93	0.752	0.895

Table 6. Computation time and memory consumption of various part of the system to produce the score of one verification trial. All tasks were performed on a 64-bit Linux server with 8G RAM and an Intel Q9550 running at 2.83GHz. All CPU times are based on one core of the processor.

Task	Task Name	CPU Time (sec.) per Utt.	% of Real Time	Memory Consumption (MB)
1	Voice Activity Detection	0.659	0.51	8.3
2	MFCC Extraction	0.095	0.25	4.2
3	Feature Warping	3.127	8.22	7.3
4	Computing Sufficient statistics	0.933	2.45	329
5	I-vector Estimation	0.817	2.15	543
6	PLDA Scoring + Tnorm/Snorm	0.012	0.03	286
	Overall	5.643	13.61	–

before cepstral mean and variance normalisation is applied. A UBM with 2048 components is used to extract iVectors with a dimension of 600. LDA is applied with dimension 600 and iVectors are centered with the subtraction of the mean obtained from the unlabelled partition of the SRE 16 development set before PLDA scoring is applied with 400 speaker factors. The UBM is learned using data from the Fisher, SRE 04-05-06 and Switchboard corpora. The T matrix and PLDA hyperparameters are learned with the same data supplemented with the SRE 08 corpus.

5. CLASSIFIER FUSION

5.1. Linear score fusion for the primary submission

Initially, 5-fold CV was performed on the dev set scores to decide on the settings (i.e. selection of base classifiers and whether to use score preprocessing or not). Pre-calibration showed systematically better results than without pre-calibration. In the selection setting, we experimented with many different ideas on how to fix the final subset of classifiers (noting that we started with 32 base classifiers). Final ensemble was selected using heuristic rule of using the best single systems ($minCprim < 0.7$) and one for each site. This resulted in better cross-validated results than using just the best best systems.

Using this setup we came up with the 17 base classifiers. Scores were first pre-calibrated using logistic regression, via minimizing the Cwlr cost, with $p_{\rm tar} = 0.01$. So scale and bias was recorded for each base classifier and applied to eval set scores. Then the linear fusion parameters were estimated on the pre-calibrated scores using the same settings.

5.2. Linear score fusion with OSCAR-penalized logistic regression weights

OSCAR-penalized logistic regression model is used to estimate weights of linear fusion of subsystem scores. Weights are obtained by fitting penalized logistic regression model in sample taken from development data, i.e. by setting Peff = 0.0075 and minimizing a cost function Cwlr in [50] subject to constraint of OSCAR method in [51]. Because of using the Cwlr cost function in [50], the output scores of the fusion can be considered as log likelihood ratios given by definition (1) in "NIST 2016 Speaker Recognition Evaluation Plan"[52].

In the simplest cases, binomial logistic regression coefficients are usually estimated by minimizing the cost function Cwlr without any constraint. However, many subsystem scores used here are highly correlated with each other, and because of this it is expected that there is much variability in values of weights among estimates based on different samples, and thus also accuracy of prediction is overly dependent on the data used for weight estimation. One way to obtain more stable estimation of weights is to constrain size of weights during the estimation process. Here the constraint of OSCAR method is used for this purpose.

OSCAR has a commonly used shrinkage method called LASSO as a special case when parameter c in [51] is set to 0. In addition to shrinking size of weights, LASSO also performs variable selection by shrinking some weights of scores to exactly zero. The problem with LASSO is that if some scores are highly correlated, LASSO often tend to reject many

of them. However, OSCAR, if c is not 0, instead of randomly selecting one variable in the group of correlated variables, tend to give correlated scores weights of the same size [53].

The shrinkage parameter λ and the tuning parameter c of OSCAR are selected by fitting several models, using different values of λ and c, to various samples of size 4000 taken from development data, and testing their prediction performance in set of validation samples of size 10000, which are taken from development in such a way that the corresponding model fitting set and the validation set are separate. The number of model fitting sets and corresponding validation sets used is 12.

Prediction performance is measured in the terms of EER, actual $C_{Primary}$ and minimum $C_{Primary}$. The weights for calculating log likelihood ratios of the evaluation data trials are chosen randomly among those 12 weights of models corresponding parameters λ and c, which produced lowest value of $C_{Primary}$ on average. However, also mean values of EER and difference between minimum $C_{Primary}$ and actual $C_{Primary}$ and their as well as $C_{Primary}$'s variation are observed.

5.3. Performance on DEV set

I4U team submitted three fusion systems – one primary and two contrastives. Performance of the fusion systems on the Dev set are shown in Table 7.

Table 7. *Performance of the primary and contrastive submissions on Dev set in terms of EER, Minimum and Actual $C_{primary}$. Each entry represents the (**Equalized** and **Unequalized**) performance metrics as defined in NIST Python scoring tool.*

Equalized	EER (%)	$C_{primary}$ (Min)	$C_{primary}$ (Act)
Primary	11.55	0.5505	0.587592
Contrastive 1	10.78	0.6565	0.796003
Contrastive 2	9.35	0.5454	0.561453
Unequalized	EER (%)	$C_{primary}$ (Min)	$C_{primary}$ (Act)
Primary	12.27	0.5280	0.558124
Contrastive 1	12.25	0.6390	0.784113
Contrastive 2	10.59	0.5053	0.524777

6. REFERENCES

[1] P. Kenny, T. Stafylakis, P. Ouellet, M.J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. IEEE ICASSP*, 2013, pp. 7649–7653.

[2] P Kenny, T. Stafylakis, P. Ouellet, and M.J. Alam, "JFA-based front ends for speaker recognition," in *Proc. IEEE ICASSP*, 2014.

[3] H. Aronowitz, "Compensating inter-dataset variability in PLDA hyper-parameters for robust speaker recognition," in *Proc. Odyssey*, 2014, pp. 282–286.

[4] N. Bümmer and E. de Villiers, "The BOSARIS Toolkit User Guide: Theory, Algorithms and Code for Binary Classifier Score Processing," Tech. Rep., AGNITIO Research, South Africa, 2011.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Prof. ICASSP*, 2014, pp. 1695–1699.

[7] K.A. Lee, H. Sun, A. Sizov, G. Wang, T.H. Nguyen, B. Ma, H. Li, W. Rao, Z.H. Lim, C. Xu, H. Xu, and X. Xiao, "Singa Submission to the 2016 NIST Speaker Recognition Evaluation," in *Proc. NIST SRE 2016*, 2016.

[8] 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. IEEE Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[9] H. Sun, B. Ma, and H. Li, "An efficient feature selection method for speaker recognition," in *Proc. ISCSLP*, 2008, pp. 181–184.

[10] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014, pp. 215–219.

[11] S. Shum, N. Dehak, R. Dehak, and J.R. Glass, "Unsupervised speaker adaptation based on the cosine similarity for text-independent speaker verification," in *Proc. Odyssey*, 2010, pp. 256–262.

[12] A. Larcher, K. A. Lee, and S. Meignier, "An extensible speaker identification SideKit in Python," in *Proc. ICASSP*, 2016, pp. 5095–5099.

[13] The Theano Development Team, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[14] 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, 2013.

[15] F. Bahmaninezhad and J. H.L. Hanesn, "i-vector/PLDA speaker recognition using support vectors with discriminant analysis," in *(submitted to) IEEE ICASSP*, 2017.

[16] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proc. IEEE ASRU*. IEEE, 2015, pp. 92–97.

[17] A. Larcher and G. Le Lan, "LIUM NIST-SRE16 SYSTEMS," in *Proc. NIST SRE 2016*, 2016.

[18] R. Saeidi, R.F. Astudillo, and D. Kolossa, "Uncertain LDA: Including observation uncertainties in discriminative transforms," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 7, pp. 1479–1488, 2016.

[19] A. Nautsch, S. Isadskiy, C. Rathgeb, and C. Busch, "HDA NIST SRE16 SYSTEMS (CRISP & I4U)," in *Proc. NIST SRE 2016*, 2016.

[20] D. Spielmann, *Spectral Graph Theory*, Chapman & Hall/CRC Computational Science. CRC Press, 2012.

[21] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, "Algorithms for graph similarity and subgraph matching," *Technical Report*, 2011.

[22] A. Shrivastava and P. Li, "A New Space for Comparing Graphs," *arXiv pre-print*, 2014, http://arxiv.org/abs/1404.4644v1.

[23] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Jensen-Bregman LogDet Divergence with Application to Efficient Similarity Search for Covariance Matrices," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 9, pp. 2161–2174, 2013.

[24] G. Liu and J.H.L. Hansen, "An investigation into back-end advancements for speaker recognition in multi-session and noisy enrollment scenarios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1978–1992, 2014.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[26] P.M. Bousquet, D. Matrouf, and J.-F. Bonastre, "Intersession compensation and scoring methods in the i-vectors space for speaker recognition.," in *Proc. Interspeech*, 2011, pp. 485–488.

[27] 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 *Proc. Odyssey*, 2012.

[28] F. Grezl, M. Karafiat, and K. Vesely, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Proc. of ICASSP 2014*, May 2014.

[29] Ulrike von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, 2007.

[30] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. 2002, pp. 849–856, MIT Press.

[31] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in *IEEE International Conference on Computer Vision*, Sept. 1999, p. 975.

[32] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data.," in *Proc. ICASSP*. Citeseer, 2013, pp. 7229–7233.

[33] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey*, Crete, Greece, Jun. 2001, pp. 213–218.

[34] T. Hasan and J.H.L. Hansen, "A study on universal background model training in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1890–1899, 2011.

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

[36] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*. ACM, 2008, pp. 1096–1103.

[37] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. ICASSP*. IEEE, 2014, pp. 4047–4051.

[38] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer, Speech and Language*, vol. 28, no. 1, pp. 295–313, Jan 2014.

[39] M. McLaren, M.I. Mandasari, and D.A. Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," in *Proc. Odyssey*, 2012, pp. 55–61.

[40] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. ICSLP*, 2006, pp. 1471–1474.

[41] C.M. Bishop, *Pattern recognition and machine learning*, springer, New York, 2006.

[42] W. Y. Chen, Y. Q. Song, H. J. Bai, C. J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 568–586, 2011.

[43] C. Kim and R.M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*. IEEE, 2012, pp. 4101–4104.

[44] J. Sohn, N.S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[45] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[46] S.J.D Prince, "Computer vision: models learning and inference," in *Cambridge University Press, 2012*.

[47] J.F. Bonastre, F. Wils, and S. Meignier, "ALIZE, a free toolkit for speaker recognition.," in *Proc. ICASSP*, 2005, pp. 737–740.

[48] 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 *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 1449–1452.

[49] H. Delgado, M. Todisco, M. Sahidullah, A. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan, "Further optimisations of constant Q cepstral processing for integrated utterance verification and text-dependent speaker verification," in *Proc. Odyssey*, 2016.

[50] V. Hautamaki, T. Kinnunen, F. Sedlak, K. A. Lee, B. Ma, and H. Li, "Sparse classifier fusion for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1622–1631, 2013.

[51] S. Petry and G. Tutz, "The OSCAR for generalized linear models," *Technical Report*, 2011.

[52] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. Leeuwen, P. Matejka, P. Schwartz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[53] H. Bondell and B. Reich, "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR," *Biometrics*, vol. 64, pp. 115–123, 2008.