May 3rd , 2010

2010 NIST Speaker Recognition Evaluation: ILPGIP (PerSay, GM, IBM Haifa, Israeli Police) System Description

Nir Krause, Gennady Karvitsky, Hagai Aronowitz, Yosef Solewicz, Ron M. Hecht

1. Overview

ILPIGB is submitting scores for the core task. In addition, PerSay is submitting scores for four other tasks. The score produced for all the tasks mentioned are obtained by different fusions of several different speaker verification systems:

- PerSay LPCC NAP SGM: An SVM-based classifier, working in the GMM model space of LPCC features, using NAP for session compensation.
- PerSay MFCC NAP SGM: An SVM-based classifier, working in the GMM model space of MFCC features, using NAP for session compensation.
- PerSay LPCC SGM: An SVM-based classifier, working in the GMM model space of LPCC features
- PerSay MFCC SGM: An SVM-based classifier, working in the GMM model space of MFCC features
- PerSay MFCC TFA: a fast scoring Total Factor Analysis model which used JFA, LDA and WCCN (based on [1]).
- GM MFCC GIB: A super-vector system that uses GIB transformation in order to reduce feature dimension.
- IBM NAP: 2 wire NAP on MFCC super vectors, as described in section 5.

The above systems used slightly different background data & parameters in each condition.

The following table shows the systems used in each condition:

Table 1: Systems used for the different conditions of NIST 2010.

Condition	ILPGIP1 (Primary)	ILPGIP 2
Core-Core	LPCC NAP SGM MFCC NAP SGM MFCC TFA GM MFCC GIB IBM	LPCC NAP SGM
core-10sec	LPCC NAP SGM MFCC NAP SGM MFCC TFA	MFCC TFA

10sec-10sec	LPCC SGM MFCC SGM MFCC TFA	MFCC TFA
core-summed	LPCC NAP SGM MFCC NAP SGM	
8summed-core	LPCC NAP SGM MFCC NAP SGM	LPCC NAP SGM

2. PerSay SGM w/o NAP systems description

2.1 Features:

- The Qualcomm-ICSI-OGI Wiener filter ([2], http://www.icsi.berkeley.edu/Speech/papers/qio/) was applied to microphone recorded segments (mic and interview).
- For phone recording energy detection over the "other" side of each 4-wire conversation was used to discard silent segments. For interview recording the ASR was used to remove the other side.
  Additional silent frames were removed by an energy-based voice activity detector with adaptive threshold.
- **LPCC:** 20 LP Cepstrum Coefficients (LPCC) + 20 delta LPCC, with mean subtraction and variance normalization, computed over 250 msec frames with 125 msec overlap.
  Or
- **MFCC:** 19 Mel Frequency Cepstrum Coefficients (MFCC) + 19 delta MFCC, including RASTA filtering, mean subtraction and variance normalization, computed over 250 msec frames with 125 msec overlap.

2.2 Super vector generation (Training & Test):

- Means-only Bayesian adaptation of the same-gender UBM, using top 10 scoring Gaussians, creates a super vector of Gaussian means. The UBM has 512 Gaussians.
- The means of the UBM Gaussians were subtracted from the target super vector.
- The means of each Gaussian were multiplied by the square root of the ratio between the Gaussian weight and the UBM Gaussian variance for this feature.
- This super vector was normalized by its L2 norm.

The relevance factor value used for most conditions is 3. For the 10sec-10sec a value of 1 was used.

2.3 Training:

The NAP-SGM systems used Nuisance Attribute Projection (NAP, [3]) to remove unwanted session & channel variability. The projection matrix was trained beforehand using data that matched the condition (see below). It was applied on the generated super vector.

2.3.1 2-wire NAP training

A more robust NAP projection was implemented by removing also directions of projection caused by other side speaker interference, as is done in [4]. In principle summed recording and it's 2 separated sides are added to the NAP which tries to find directions to project away. The 2 wire projection dimension was 5 for males and 40 for females.

An SVM was trained for each training file, using as features the super vector obtained by the previous steps.

Additional details on SVM classification in the GMM model space can be found in [5]. The SVM classifier was implemented in SVMTorch [6], using a first-degree polynomial kernel.

2.3.2 8-summed training

8 summed training was done by first automatically separating each call to its 2 sides by [7]. We get 16 files. We train for each a super vector (after NAP) and an SVM model. Then we score with each SVM model all the files. On this matrix we define the neighbors of each file as those who got scores > -0.9. We take a super vector to create the speaker model if it has at least 5 neighbors, and its pair (from the original conversation) has less than 5. Using all the chosen super vectors we create an SVM model.
We do a 2^{nd} pass with this model –We choose from each pair of super vectors the one with the highest score with regards to the SVM model, given its score is above -0.9. The chosen files are used to create a new SVM model. Usually this model includes 7 to 8 files.

2.4 Testing:

The trained SVM classifier was used to classify the test super vector, and output its margin, as the score.

2.4.1 Summed audio testing:

When the test segment was summed (2-wire), an external segmentation utility was used to divide the test segment to two sides. The external segmentation utility [7] is using two feature sets (FFT-based spectrum and LPCC) and two self-organizing–maps (SOM) classifiers in an iterative fashion to cut the summed file into two files which presumably hold the voice of only one speaker. Each such file was tested against a model created from the train segment. The highest score was selected as the score of this train-test pair.

2.5 Parameters and background data

The background model the system includes English audio from SRE 99, 03, 04, 05, 06, and 08.
The 2008 SRE was partitioned. Speakers with many calls were used to train NAP. Speakers with a few calls were used to create a test set.
In the core-10 seconds, the training audio was cut to 15 seconds pieces, to match the test audio length.
NAP dimension was 20 for 10 seconds conditions, and between 50 to 200 in other cases.

2.6 Execution times

Wiener Filter: ~20 sec for each segment
Preprocessing LPCC: ~ 1.5 sec for each segment, including feature extraction, Baum Welch statistics computation & super vectors generation
Preprocessing MFCC: ~ 4 sec for each segment, including feature extraction & super vectors generation
NAP projection: ~ 0.13 sec for each segment
SVM training: ~ 1.2 sec for each segment
SVM testing: ~ 0.02 sec for each segment
Summed file separation: ~ 30 sec for each summed segment

The execution times are per one 5 min segment.
Processing was done on an Intel P4 with 4GB memory, running Linux.

3. PerSay Joint Factor Analysis with Fast Scoring System

Joint Factor Analysis (JFA) with simplified scoring based on cosines kernel. Channel normalization based on LDA+WCCN ([1]). The total variability model dimension is 150, reduced by LDA to 120. The JFA model and LDA/WCCN computed on two different subsets of FISHER + NIST04-08 data. The underlying feature set is 19MEL+19deltas. The final scores are normalized by znorm, with the znorm impostors taken from NIST04,05,06. The size of impostor set for znorm normalization is approximately 500 speakers.

3.1 Execution times

Training: ~1.2 sec for each 10sec segment, ~5 sec for all others
Testing: ~1.2 sec for each 10sec segment, ~5 sec for all others
Processing was done on an Intel 2140@1.60GHz with 4G memory, running Linux.

4. GM MFCC GIB System

One of the important challenges for super-vector systems [8],[9] is the super-vector dimension reduction. The goal of this challenge is to find a representation that is both compact and effective. In this system the dimension reduction procedure was based on the Information bottleneck (IB) [10] and more specifically on the Gaussian Information Bottleneck (GIB) [11]. This GIB evaluation system is based on [12].

4.1 Experiment:

Several sets of experiments were conducted on the NIST 2008 data. Those experiments were not conducted on the regular evaluation sets. A different partition of the conversations was made such that it better reflects the NIST 2010 scenarios. A total of four scenarios were defined according to the evaluation conditions:

-All trials involving interview speech from the same microphone in the training and test. This scenario is denoted as int-int-same.

-All trials involving interview speech from the different microphone in the training and test. This scenario is denoted as int-int-diff.

-All trials involving telephone speech in the training and test. This scenario is denoted as phn-phn.

-All trials involving interview speech for training and telephone speech for test. This scenario is denoted as int-phn.

The scores were TZ normalized by a set of speakers that were omitted from the scenarios described above. A total of 3000 segments were used for the normalization. Those segments were used to estimate the GIB transformation as well.

4.2 Results:

Results of the different scenarios on the NIST 2008 are shown in the following table.

Table 2: NIST 2008 recognition results (equal error rate) for different scenarios and genders. SV – represents super-vector system without dimension reduction. GIV- represents super-vector system with GIB based dimension reduction.

Gender	scenario	SV(EER)	GIB(EER)
Female	int int diff	17.7	7.7
Female	int int same	6.8	4.3
Female	phn phn	6.9	3.5
Female	Int phn	20.3	11.8
Male	int int diff	14.4	8.5
Male	int int same	4.7	4.2
Male	phn phn	7.3	6.6
Male	Int phn	18.6	10.6

The GIB proves itself as an effective tool for extraction of relevant information.

5. IBM System

5.1 Features:

- Adaptive energy based voice activity detection
- MFCC features + deltas (26 in total)
- feature warping

5.2 Model:
- GMM supervector parameterization of all sessions (enrolment, development, test) estimated using relevance-MAP
- NAP compensation using 2-wire variant described in [4].
- ZTnorm score normalization using unbiased scoring [13]
- H-norm for normalization of tel-tel vs. mic-mic scores

5.3 Datasets used:
NIST 04 - UBM training, NAP estimation
NIST 05 - ZT-norm, H-norm
NIST 06 - NAP estimation, ZT-norm, H-norm
NIST 08* ZT-norm, H-norm
* speakers not in ILNist dev. dataset

6. Fusion & Score calibration

The scores augmented by side-information based on the length of the respective utterances were fused using the Focal toolkit with[14], with the linear logistic regression algorithm.

7. References

[1] Dehak, N., Dehak, R., Kenny, P., Brummer, N., Ouellet, P and Dumouchel, P., "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification", in Proc. of Interspeech, 2009.
[2] Adami A. et. al, "Qualcomm-ICSI-OGI Features for ASR", in Proc. of ICSLP2002.
[3] Solomonoff A. et al., "Advances in channel compensation for SVM speaker recognition", in Proc. of ICASSP2005.
[4] Solewicz Y. and Aronowitz H., "Two-Wire Nuisance Attribute Projection", in Proc. of Interspeech, 2009.
[5] Krause N. and Gazit R., "SVM-based Speaker Classification in the GMM Models Space", in Proc. Odyssey 2006
[6] Collobert R., Bengio S., Mariéthoz J., "Torch: a modular machine learning software library", Technical Report IDIAP-RR 02-46, IDIAP, 2002.
[7] Metzger Y., "Blind Segmentation of a Multi-Speaker Conversation Using Two Different Sets of Features", in Proc. of Odyssey 2001
[8] Aronowitz, H., Burshtein, D. and Amir, A., "Speaker Indexing in Audio Archives Using Test Utterance Gaussian Mixture Modeling", in Proc. of ICSLP, 2004.
[9] Campbell, W. M., "Generalized Linear Discriminant Sequence Kernels for Speaker Recognition ", in Proc. of ICASSP, 2002.
[10] Tishby, N., Pereira, F., and Bialek W., ``The Information Bottleneck Method", The 37th annual Allerton Conference on Communication, Control, and Computing, 1999.
[11] Chechik, G., Globerson, A., Weiss, Y. and Tishby, N., "Information Bottleneck for Gaussian Variables", Journal of Machine Learning Research (JMLR) 6:165-188,2005.
[12] Hecht, R. M., Noor, E. and Tishby, N., "Speaker Recognition by Information Bottleneck", in Proc. of Interspeech, 2009.

[13] Aronowitz H., "Efficient Score Normalization for Speaker Recognition", in Proc. of ICASSP 2010.
[14] Brummer N. and du Preez J., “Application-Independent Evaluation of Speaker Detection”, *Computer Speech and Language*, 2005