SYSTEM DESCRIPTION

Institute for Infocomm Research (IIR) site
Contact person: Haizhou LI, E-mail: hli@i2r.a-star.edu.sg
May 16, 2006

1. Introduction

This document describes the speaker recognition systems from the Institute for Infocomm Research (IIR) site. The systems are built to participate in NIST 2006 Speaker Recognition Evaluation (SRE). Institute for Infocomm Research site submits results from three systems, which are kept in three files:

1. IIR_1.txt
2. IIR_2.txt
3. IIR_3.txt

We built 6 subsystems and used ensemble method to fuse the outputs from 6 individual subsystems, namely,

1. Spectral SVM-MFCC
2. Spectral SVM-LPCC
3. MFCC-GMM-UBM
4. TDCT-GMM-UBM
5. Bag-of-Sounds
6. GMM Tokenization

Each subsystem is an individual classifier. An ensemble of classifiers is a set of classifiers whose individual scores are combined in the classification process. The combined score is used as the final output and to make True/False decision, as reported in the results files.

We take 2 strategies to fuse the outputs from the 6 subsystems, leading to Submission 1 and Submission 2. Submission 3 is the output from Spectral SVM-LPCC. Submission 1 represents the primary system.

IIR site submits results under 7 task conditions as shaded in Table 1. The confidence scores are to be interpreted as likelihood ratios.

		Test segment condition			
		10sec 2chan	1conv 2chan	1conv summed chan	1 conv aux mic
Training condition	10sec 2chan	10sec4w-10sec4w			
	1conv 2chan	1conv4w-10sec4w	1conv4w-1conv4w		
	3conv 2chan	3conv4w-10sec4w	3conv4w-1conv4w		
	8conv 2chan	8conv4w-10sec4w	8conv4w-1conv4w		
	3conv summed chan				

Table 1. IIR site participating task conditions

Next, we briefly describe the 6 subsystems in Section 2 to Section 5. In Section 6 to 8, we describe the 3 submissions.

2. Spectral SVM-MFCC & SVM-LPCC subsystems

The spectral SVM system is based on the work reported in [1], [2], [3]. The feature vectors (with a dimension of 36) extracted from an utterance are expanded to a higher dimensional space by calculating all the monomials. All monomials up to order 3 are used, resulting in a feature space expansion from 36 to 9139 in dimension. The expanded features are then averaged to form an average expanded feature vector for each of the utterances under consideration. In the implementation, it is also assumed that the kernel inner product matrix is diagonal for computational simplicity.

During enrollment, the current speaker under training is labeled as class +1, whereas a target value of -1 is used for the background speakers. The set of background data is selected from Switchboard 3 Phase 1 and 2 (for Cellular data) and Switchboard 2 Phase 2 and 3 (for landline telephone), 4 datasets. We randomly select 2000 utterances from each of the 4 datasets to form a background speaker database of 8000 utterances, with roughly equal amounts (4000 utterances) from male and female speakers. For each utterance in the background and for the current speaker under training, an average expanded feature is created. These average expanded features (assigned with appropriate label) are used in the SVM training. The commonly available SVMTorch [4] is used for this purpose. The result of the training is a vector $\mathbf{w}$ of dimension 9139 which represents the desired target speaker model [3]. During evaluation, an average expanded feature vector $\mathbf{b}$ is formed for each of the input utterances, and the score is taken as the inner product between these two vectors, i.e., $\mathbf{w}^T\mathbf{b}$.

Two different sets of acoustic spectral features, namely mel-frequency cepstral coefficients (MFCC) and linear prediction coding coefficients (LPCC), both with a dimension of 36, are used thereby forming two separate SVM systems. For the MFCC front-end, we use a 27-channel filterbank, and 12MFCC + 12Δ + 12ΔΔ coefficients. On the other hand, 18LPCC + 18Δ coefficients are used for the LPCC front-end. A standard voice activity detection (VAD) is applied after feature extraction. Mean subtraction and variance normalization are applied for the detected speech frames, for both the MFCC and LPCC features.

Test normalization (Tnorm) method is used to normalize the score [5]. The NIST 2004 training data is used to form the cohort models. In particular, the speaker models in the NIST 2004 are used as the cohort models. By so doing, the training condition for the cohort models can be match to that of the target speaker models. For example, the trained models in the 1side of NIST 2004 are used as the cohort models for the

target models in the 1conv4w training condition of the NIST 2006. Similar concept is applied to 10sec4w, 3conv4w, and 8conv4w training conditions.

3. MFCC-GMM-UBM and TDCT-GMM-UBM subsystems

Our GMM-UBM system uses the standard set-up described in [6]. We have two separate GMM-UBM subsystems. The first one is based on MFCC, whereas the second one uses a new feature set termed temporal discrete cosine transform (TDCT) features [7]. The TDCT features are derived from the MFCC features by computing discrete cosine transform over several frames, spanning temporal context of about 250 milliseconds. The lowest DCT coefficients of each MFCC stream are stacked to form a long vector (108 dimensions). Similar mean subtraction and variance normalization is also applied for the detected speech frames (after VAD) for the TDCT features.
The background models are trained from the NIST2004 1-side training data subset. We train gender-dependent background with 256 Gaussians in each model. In the training phase, the background model having the same gender with the target is used for adaptation. In the scoring phase, the same-gender background model with the target is used to give the background score.

4. Bag-of-Sounds subsystem

This approach uses phoneme tokenizers as the front end, and vector space modeling as the back end classifier [8]. For a speech utterance, the phoneme tokenizers generate phone sequences, which are then converted to a vector of weighted terms. The vector is compared with a speaker's SVM model and a score is produced. NIST SRE 2002 corpus is used as background data in our experiment.

Seven phoneme tokenizers are used in our system: English, Korean, Mandarin, Japanese, Hindi, Spanish and German. English phonemes are trained from IIR-LID database [9]. Korean phonemes are trained from LDC Korean corpus (LDC2003S03). Mandarin phonemes are trained from MAT corpus [10]. Other phonemes are trained from OGI-TS corpus [11]. We use 39-dimensional MFCC features. Utterance based cepstral mean subtraction is applied to the MFCC features to remove channel distortion. Each phoneme is modeled using a three-state HMM. The English, Korean and Mandarin states are of 32 mixtures each, while others are of 6 mixtures considering the availability of training data. Phoneme recognition is performed with a Viterbi search using a fully connected null-grammar network of phones.

For a speech utterance, the tokenizers yield seven phoneme sequences. They are converted to a vector of weighted terms in three steps. Firstly, we compute unigram and bigram probabilities for each phoneme sequence, and then concatenate the probabilities into a vector. Secondly, each entry in the vector is multiplied by a background component. We adopt the log-likelihood ratio weighting [12]. Finally, we concatenate the seven vectors to form a long vector.

In the training process of the SVM, each conversation side in the corpus is treated as a "document". A single vector of weighted probabilities is derived from a conversation side. We use a one-versus-all strategy to train a model for a given speaker. The speaker's conversations are trained to a SVM target value of +1. All conversation sides in the background corpus are used as the class for SVM target value of -1. In the test process, the vector of the input speech is introduced into a speaker's SVM model and a score is produced. This score is compared to a threshold and a reject or accept decision is made based upon whether the score is below or above the threshold. The SVMTorch package [4] with a linear kernel is used in our experiment. Training is performed with the parameter setting of c=1.

5. GMM Tokenization subsystem

This approach uses multiple GMM tokenizers as the front end, and vector space modeling as the back end classifier [13]. Each GMM tokenizer converts the input speech into a sequence of GMM token symbols which are indexes of the Gaussian components scoring highest at every frame in the GMM computation. The GMM token sequences are then processed in the same way as the process of phone sequences in the bag-of-sounds approach, i.e., the sequences are converted to a vector of weighted terms and then recognized by a speaker's SVM model.

Inspired by the finding of PPRLM in language recognition where multiple parallel single-language phone recognizers in the front-end enhance the language coverage and improve the language recognition accuracy over single phone recognizer, we explore multiple GMM tokenizers to improve speaker characteristics coverage and to provide more discriminative information for speaker recognition [13]. By clustering all the speakers in the training set into several speaker clusters, we represent the training space in several partitions. Each partition of speech data can then be used to train a GMM tokenizer. After the multiple parallel GMM tokenizers are constructed, a speech segment passes through all these tokenizers to be converted into multiple feature vectors, which are then concatenated to form a composite vector. We use the NIST SRE 2002 corpus for the training of speaker cluster based GMM tokenizers, and use the NIST SRE 2004 corpus as the background data. 10 parallel GMM tokenizers, each having 128 mixture of Gaussian components, are constructed

6. System for Submission 1 (Primary)

For a given speech segment and its reference speaker model, a 6 dimensional score vector are derived from the 6 subsystems, namely Spectral SVM-MFCC, Spectral SVM-LPCC, MFCC-GMM-UBM, TDCT-GMM-UBM, Bag-of-sounds and GMM Tokenization. We used NIST SRE 2005 evaluation corpus as the development data for classifier fusion. All the score vectors from the same trial condition are used as training data for that trial condition.

The final classifier is a SVM classifier. We apply polynomial expansion of up to order 3 on the 6 dimensional score vectors before presenting the vectors to linear SVM. 3 SVMs are trained for order=1, order=2 and order=3, the output score is the average of these three SVM outputs.

We use the NIST 2005 evaluation corpus to obtain the threshold and make True/False decision for submission 1.

7. System for Submission 2

In Submission 2, we take the same strategy as in Submission 1 to fuse the scores from 6 subsystems to obtain the final

score. However, for True/False decision making, we make decision for each individual subsystem based on thresholds that are obtained on NIST 2005 evaluation corpus. As a result, each subsystem has a True/False decision for each test utterance. We use majority vote to make collective decision for Submission 2.

8. System for Submission 3

Submission 3 is the results from Spectral SVM-LPCC.

9. CPU Execution Time

The CPU time for training and testing of ensemble classifier is negligible as compared with the training and testing of the subsystems. In Table 2, we report the CPU time required by each of the subsystems. As the training data for UBM vary from system to system, we report the training time requirement of UBM in terms of absolute hours and that of the training of target speaker model and testing in xRT on an Intel Xeon 2.8GHz CPU with 1GB memory.

Subsystems		Training-test conditions	Training		Test (xRT)
			UBM (hours)	Speaker (xRT)	
	MFCC-GMM-UBM	1conv4w-1conv4w	24	0.017	0.017
		1conv4w-10sec4w		0.017	0.050
		3conv4w-1conv4w		0.016	0.017
		3conv4w-10sec4w		0.016	0.050
		8conv4w-1conv4w		0.016	0.017
		8conv4w-10sec4w		0.016	0.050
		10sec4w-10sec4w		0.020	0.050
	TDCT-GMM-UBM	1conv4w-1conv4w	24	0.027	0.033
		1conv4w-10sec4w		0.027	0.100
		3conv4w-1conv4w		0.030	0.033
		3conv4w-10sec4w		0.030	0.100
		8conv4w-1conv4w		0.030	0.033
		8conv4w-10sec4w		0.030	0.100
		10sec4w-10sec4w		0.040	0.100
	Spectral SVM-MFCC	all 7 task conditions	N.A.	0.700	0.015
	Spectral SVM-LPCC	all 7 task conditions	N.A.	0.700	0.015
	Bag of Sounds	all 7 task conditions	35	2.400	0.400
	GMM Tokenization	all 7 task conditions	105	1.040	0.100

Table 2. CPU execution time requirements of subsystems.

References

[1] W. M. Campbell, "A sequence kernel and its application to speaker recognition," in *Proc. NIPS 2001*.

[2] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP*, pp. 161-164, 2002.

[3] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support Vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210-229, 2006.

[4] R. Collobert and S. Bengio, "SVMTorch: support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143-160, 2001.

[5] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no 1-3, pp. 42-54, Jan 2000.

[6] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, 10(1):19--41,2000.

[7] Tomi Kinnunen, Chin Wei Eugene Koh, Lei Wang, Haizhou Li, Eng Siong Chng, "Shifted Delta Cepstrum and Temporal Discrete Cosine Transform Features in Speaker Verification", submitted to *Interspeech 2006.*

[8] H. Li and B. Ma, "A Phonotactic Language Model for Spoken Language Identification", *43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*, June 2005, Ann Arbor, USA.

[9] Language Identification Corpus of the Institute for Infocomm Research

[10] H.-C. Wang, "MAT-a project to collect Mandarin speech data through networks in Taiwan," *International Journal of Computational Linguistics Chinese Language Processing*. 1 (2) (February 1997) 73-89.

[11] http://cslu.cse.ogi.edu/corpora/corpCurrent.html

[12] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Proc NIPS 2003*.

[13] B. Ma, D. Zhu, R. Tong and H. Li, "Speaker cluster based GMM tokenization for speaker recognition," submitted to *Interspeech 2006.*