SYSTEM DESCRIPTION

Institute for Infocomm Research and University of Joensuu (IIR-JoY) joint submission
Contact person: Dr Tomi Kinnunen, E-mail: ktomi@i2r.a-star.edu.sg

May 16, 2006

1. Introduction

This document describes the speaker recognition systems from the joint submission of Institute for Infocomm Research (IIR) and University of Joensuu (IIR-JoY). The systems are built to participate in NIST 2006 Speaker Recognition Evaluation (SRE). We submit results from three systems, which are kept in three files:

1. IIR-JoY_1.txt
2. IIR-JoY_2.txt
3. IIR-JoY_3.txt

The confidence scores in the files can be interpreted as likelihood ratios. Each of the submissions uses three subsystems based on spectral (SVM-LPCC), prosodic (Long-term F0), and symbolic (GMM tokenization) information. A second-level classifier combines the scores of the three subsystems. We take three approaches for the second-level classifier, including neural network (NN), support vector machine (SVM), and the combination of these (NN+SVM). These comprise the following three systems from IIR-JoY:

- Submission 1: (**primary system**): NN+SVM combiner
- Submission 2: NN combiner
- Submission 3: SVM combiner

We submit results under the 7 task conditions highlighted in Table 1.

		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-JoY participation task conditions

2. SVM-LPCC subsystem

The spectral SVM system is based on the work reported in [1], [2], [3]. The front-end of the system uses 18LPCC + 18Δ coefficients (36-dimensional vectors). A standard voice activity detection (VAD) is applied after feature extraction. Mean subtraction and variance normalization is applied for the detected speech frames.

The feature vectors are expanded to a higher dimensional space by 3rd order polynomial expansion, resulting in a new feature space of 9139 dimensions. The expanded features are then averaged to form an average expanded feature vector for each of the utterances under consideration.

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 speaker model is a weight vector [3] of dimension 9139. For the test utterance, average expanded feature of the same dimensions is computed and the similarity score is given by the inner product between the model vector and the unknown speaker vector.

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 applied to 10sec, 3conv4w, and 8conv4w training conditions.

3. Long-Term F0 Distribution Subsystem

This approach is based on comparing long-term fundamental frequency (F0) statistics between the training sample and the test sample [8]. The fundamental frequency is estimated using the YIN method [9], and F0 is represented in log scale. Both the training sample and the unknown sample are converted into histograms of 63 bins, and the histograms are compared by evaluating the Kullback-Leibler divergence between them.

4. GMM Tokenization subsystem

This approach uses multiple GMM tokenizers as the front end, and vector space modeling as the back end classifier [7]. 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 converted to a vector of weighted terms and then recognized by a speaker's SVM model [6].

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 [7]. 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 mixtures of Gaussian components, are constructed

For a speech utterance, the tokenizers yield ten GMM token sequences. They are converted to a vector of weighted terms in three steps. Firstly, we compute unigram and bigram probabilities for each GMM token 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 [6]. 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. System for Submission 1 (Primary)

For a given test segment – claimed model pair, a 3-dimensional score vector is produced by the three subsystems (SVM-LPCC, Long-term F0, GMM tokenization). The three subsystem scores are combined into a single score by using a combination of neural network (NN) and a support vector machine (SVM).

For the neural network, we use multilayer perceptron with sigmoid activation functions and single output. For the SVM, we apply polynomial expansion of up to order 3 on the 3-dimensional score vectors before presenting the vectors to linear SVM. Three SVMs are trained for order=1, order=2 and order=3, the output score is the average of these three SVM outputs. The final score is obtained as the average of the neural network and SVM outputs.

We used NIST SRE 2005 evaluation corpus as the development data for the score combiners. All the score vectors from the same trial condition are used as training data for that trial condition. We use the NIST 2005 evaluation corpus to obtain the threshold and make True/False decision.

6. System for Submission 2

Submission 2 is similar to the primary submission, but uses only neural network for the score fusion.

7. System for Submission 3

Submission 3 is similar to the primary submission, but uses only SVM for the score fusion.

8. 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 the three subsystems. The training time requirement of UBM is given 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.

	Training		Test (xRT)
	Background model (hours)	Speaker (xRT)	
SVM-LPCC	N.A.	0.700	0.015
Long-term F0	N.A.	0.040	0.040
GMM Tokenization	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] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," *Proc NIPS,* 2003.

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

[8] T. Kinnunen and R. Gonzalez-Hautamäki, “Long-Term F0 Modeling for Text-Independent Speaker Recognition”, *Int. Conf. on Speech and Computer (SPECOM'2005)*, Patras, Greece, 567-570, October 2005.

[9] A. de Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", *Journal of the Acoustical Society of America*, vol. 111, nro. 4, April, 2002.