System description of Voice_of_R

Zeng Hui[1], *Jiang Feiying*[2]

[1] Beijing ROOBO Technology Co., Ltd
[2] Beijing ROOBO Technology Co., Ltd
zenghui@roobo.com, jiangfeiying@roobo.com

Abstract

Our team took part in task2 and task3 in FFSVC2020. For both tasks, we combine I-vector with X-vector to do speaker verification. The system is implemented based on Kaldi toolkit. The method of I-vector uses a 2048 Gaussian to describe the universal background model (UBM), after doing adaption for a specific speaker, convert the GMM supervector to a 400-dimensions I-vector to represent the voice biometry of this speaker and scoring by PLDA. The method of X-vector system uses a TDNN network to generate speech embedding, same as the I-vector, also use PLDA to do soring of embedding. Finally, we use the mean of these two PLDA scores to represent the final score of a test. The ERR of our test result for task 2 is 11.5% and for task 3 is 10.08%.

Index Terms: speaker recognition, voice biometrics, computational paralinguistics

1. Introduction

The traditional voice biometrics technique is based on the GMM-UBM model. In this framework, the target model of a speaker is adapted from the UBM, this adapted model contains both the information of channel and the speaker. But there is bad effect on the speaker recognition due to the channel interference, to decrease the impact of the channel interference to get better performance. An effective way should be used to map all the information into a compact vector, which can help to distinguish the channel and speaker from the original data. Inspired by the JFA theory, [1] uses a compact representation called I-vector to summarize the identity feature of a speaker utterance. Given a speech recording h, normally, the UBM and the adapted model is represented by supervector, which is created by stacking all mean vectors from GMM. The I-vector is described by following.

$$M_{s,h} = m_u + Tw_{s.h} \quad (1)$$

m_u a UBM supervector, it's not relevant to any specific speaker;

$M_{s,h}$ the target speaker supervector;

T a total variability space.

Use the EM algorithm to train the T matrix and the $w_{s.h}$ can be extracted from (1). Therefore, the I-vector contains channel and speaker info. And then PLDA [2] is applied to distinguish these two signals from I-vector, to reduce the effect of different channel. The PLDA is modeled as following

$$X_{ij} = \mu + Fh_i + Gw_{ij} + \varepsilon_{ij} \quad (2)$$

$\mu + Fh_i$ is the speaker info part, describes the difference between different persons (the difference of inter-class).

- μ is the training data mean
- *F is a matrix, every column is a feather to describe the inter-class*
- *h_i is a specific feature of a person*

$Gw_{ij} + \varepsilon_{ij}$ is the noisy part, describes the difference between different utterances from one person (the difference of intra-class)

- *G is a matrix, every column is the feature of intra-class*
- *w_{ij} is a specific feature of specific speech recording of a person*
- *ε_{ij} is the error of unknown.*

Same as training the T space in I-vector, EM algorithm is apply to train the parameters μ, F, G and Σ. Finished training PLDA, following scoring method is used to evaluate performance:

$$S_{cl} = \log \frac{P(W_{cl}, W_{tst}|H_0)}{P(H_{cl}|H_1)P(W_{cl}|H_1)} \quad (3)$$

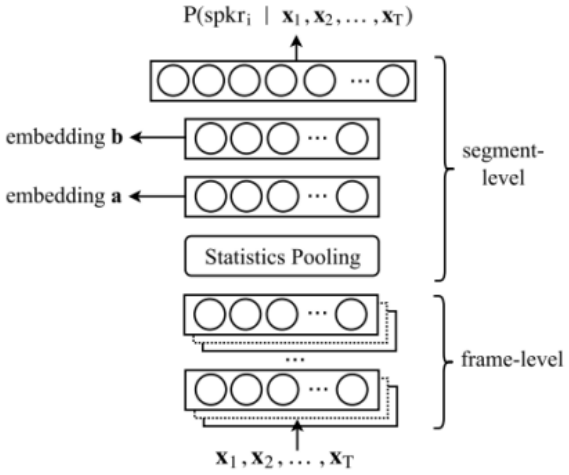
W_{cl} is the I-vector of the target speaker;

W_{tst} is the I-vector corresponding to the test utterance;

H_0 is the same speaker hypothesis;

H_1 is the hypothesis that the target and test I-vectors are from different speakers.

Since the nature network is developed very fast, [3] proposed X-vector to extract embedding feature from the TDNN. The network is as following.



The first five layers of time-delay network are frame-level, The sixth layer is a statistics pooling layer, which is responsible for calculate the mean and standard deviation of frame-level and map it to segment-level layer. After getting the embedding,

also use PLDA to do speaker classification. If there are sufficient data, X-vector can get better performance. Table 1 is the detail of input dimension and context of every level.

Table 1: *The embedding DNN architecture*

Layer	Layer context	Total context	Input x output
frame1	[t – 2, t + 2]	5	120 x 512
frame2	{t – 2, t, t + 2}	9	1536 x 512
frame3	{t – 3, t, t + 3}	15	1536 x 512
frame4	{t}	15	512 x 512
frame5	{t}	15	512 x 1500
stats pooling	[0, T]	T	1500T x 3000
segment6	{0}	T	3000 x 512
segment7	{0}	T	512 x 512
softmax	{0}	T	512 x 4430

2. Data description

Besides the data from the ffscv2020 challenge dataset, it's the in-domain (ID) data, we also download dataset from OPENSLR website, including the Aishell data (SLR33), Datatang data (SLR62) and the HI MIA data (SLR85), they are the out-of-domain (OOD) data. To train X-vector system, we need to augment these data amount. The detail of data description is showed in table 2

Table 2: *data description*

Data type	number
OOD Clear data	1.3 million
ID clear data	520 thousand
OOD noisy data	650 thousand
ID noisy data	260 thousand

2.1. Add Noisy

We use a MATLAB program to add different noisy to the original data with different SNR. The noisy includes light music, rock music, car noisy and office noisy, and the SNR is from 10db to 20db. The proportion of these noisy data is: pick 30% data to add music noisy, 10% data to add car noisy and 10% data to add office noisy. The finally proportion of the clean and noisy data is 2:1.

2.2. Speech Signal Enhancement

Most part of the challenge dataset and the HI-MIA dataset are recorded by the microphone array, we can use the different phrase info to do signal enhancement to get clearer speaker voice signal.

For the tasks, we use the signal processed by the enhancement program to do enrollment and verification.

2.3. VAD

Use the KALDI tool sid/compute_vad_decision.sh to do VAD. And the noisy data shared the same VAD result with the original data.

2.4. Down sampling

Experiment shows using different frequency to train model, the performance is not good. The audio data recorded by the phone is 48000Hz, since the frequency of audio recorded by the microphone array is 16000Hz, use sox to down sampling the frequency of phone audio to 16000Hz,

3. The system

We combine I-vector and X-vector to do speaker recognition. Both methods process audio with down sampling and signal enhancement, then extract 20-dim MFCC applying CMVN as feature input and apply KALDI VAD.

3.1. I-vector

The I-vector part follow the description in section 1 to train the UBM with 2048 GMM, the UBM model size is 14.7MB. After training T space, the 400-dim I-vector can be extracted to do PLDA training, the PLDA model size is 1.22MB. Since with limit data I-vector can got a standard performance, we only use the clean data from Aishell and Datatang to train the UBM, and use clean data of this year's challenge to get T space, and calculate I-vector to train PLDA. The final test use only once data to do enrolment, no data augment is apply.

3.2. X-vector

The X-vector system is following the architecture of [3], Use the clean and noisy data from the Aishell, Datatang and HI_MIA, which last more than 1 second, to train the TDNN, the final model size is 25.7 MB. There are totally 4430 speakers for training, therefore finally a 4430-dim embedding is got. Use the clean and noisy data from this year to get speech embedding from the TDNN, and train a LDA to reduce the embedding dimension to 150. Finally, the PLDA matrix is train by these in-domain 150-dim vectors, the PLDA model size is 178KB.

3.3. Test Result

Use the mean of scoring from I-vector and X-vector to represent the final score. The ERR of our test result is in table 3.

Table 3: *test result*

test	ERR
Task 2 dev set	12%
Task 3 dev set	10.4167%
Task 2 test set	11.5%
Task 3 test set	10.08%

4. Conclusions

The far field data provided by the FFSVC2020 is a good dataset for us to do experiment to investigate the different channel interference impact on the speaker verification. We apply the signal enhancement technique to get more robust audio signal for speaker verification. And combine I-vector system with X-vector system to do experiment to get familiar with the popular technique used in speaker recognition area.

5. References

[1] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, R.Dehak, "Language Recognition via Ivectors and Dimensionality Reduction," *INTERSPEECH 2011*.

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

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2018.