THE THU-SPMI SRE-16 SYSTEM WITH JOINT BAYESIAN SCORING AND LADDER NETWORK BASED FEATURE LEARNING

Yiyan Wang, Haotian Xu, Zhijian Ou

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, Beijing, China
wangyiya14@mails.tsinghua.edu.cn, xht13@mails.tsinghua.edu.cn, ozj@tsinghua.edu.cn

1. INTRODUCTION

We submit three different systems to participate in the fixed training condition. We designate System 3 as the primary system.

The distinctive features of our systems are the novel use of Joint Bayesian scoring and ladder network based feature learning. Due to limited time, we use a simple front-end - the DNN used for i-vector extractor is trained using only clean data. Multi-condition training should greatly improve the front-end.

2. SUBMITTED SYSTEMS

Our speaker recognition system consists of three main modules, which are i-vector extractor based on Deep Neural Networks (DNNs), i-vector post-processing (length normalization) and discriminant analysis. We submit three different systems which are described briefly as follows. The system flowchart is shown in Figure 1, and the training data statistics in system building are summarized in Table 1. The performance of the three systems on the SRE 2016 ”dev” set (i.e. LDC2016E46-SRE16-CallMyNet-Training-Data labeled part) is reported in Table 3. It can be seen that the combined System 3 gives the best performance on this ”dev” set.

2.1. System 1

2.1.1. *i-vector extractor and post-processing*

System 1 uses DNN-based i-vector extractor. The acoustic features used are 40-dimensional Mel-Frequency Cepstral Coefficients (MFCCs), including 20-dimensional static features and first-order derivatives. A DNN is trained on FISHER data with 5 hidden layers and 5335 senones. The input of the DNN is the MFCCs extracted using 21 frames (11 frames before and 9 frames after). The i-vector dimension is 600 and the i-vector extractor is trained by the EM algorithm. The extracted i-vectors are post-processed by length normalization to $sqrt(600)$. The above steps are mostly conducted using Kaldi toolkit [1].

2.1.2. *Discriminant analysis: Joint Bayesian*

Joint Bayesian (JB) is used in our system for discriminant analysis, which is originally proposed in [2] for face verification and further developed in [3] for speaker verification. In JB, i-vectors are modeled as two independent Gaussians which represent speaker identity and intersession residuals respectively. The j-th i-vector of speaker i, denoted by $x_{ij} \in R^d$, is decomposed as:

$$x_{ij} = \mu_i + \varepsilon_{ij} \tag{1}$$

where $\mu_i \sim \mathcal{N}(0, S_\mu)$ is the speaker identity variable, $\varepsilon_{ij} \sim \mathcal{N}(0, S_\varepsilon)$ models the within-speaker variability. The model parameters are $\Theta = \{S_\mu, S_\varepsilon\}$ which is estimated by the EM algorithm through iteratively optimizing the expected complete log-likelihood function.

In speaker verification testing, we calculate the log-likelihood ratio (LLR) as the score to determine whether one set of enroll i-vectors x_1 (including one or three i-vectors) and the test i-vector x_2 are from the same speaker

$$LLR(x_1, x_2) = logp(x_1, x_2) - logp(x_1) - logp(x_2) \tag{2}$$

2.2. System 2

System 2 inherits the i-vector extractor module from System 1. In System 2, we propose to use ladder networks [4, 5] with center loss [6] to non-linearly extract 500-dim features from the 600-dim i-vectors generated in System 1. Then Fisher LDA is applied, which makes the features more like Gaussians. Finally, JB-based scoring is used to generate the verification score as in Equ. (2).

The ladder networks [5] are initially designed for unsupervised feature learning and are extended in [4] for semi-supervised feature learning. For our System 2, we train a ladder network using both speaker labeled i-vectors (Switchboard-2 and SRE which basically are out-of-domain data for evaluation) and un-labeled i-vectors (a subset from LDC2016E46-SRE16-CallMyNet-Training-Data unlabeled part which are more matched to the evaluation data). The motivation is to leverage unlabeled in-domain data.

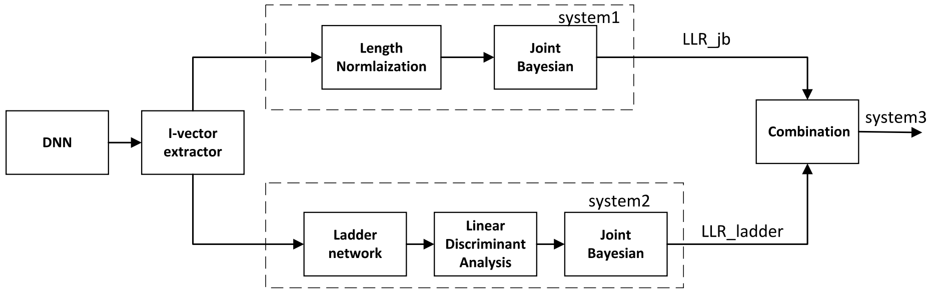

Fig. 1. The flowchart of our speaker verification systems.

	Dataset	Speaker number	Segment number	Duration (hours)
DNN	FISHER	4712	23387	600
i-vector extractor	SRE, SWBD, FISHER	14681	80909	3900
JB	SRE	3805	36612	1800
Ladder network training	SRE, Switchboard-2	8039	109667	2400

Table 1. The statistics of training datasets for different modules in system building. SRE denotes the SRE 04 05 06 08 evaluation data collection. SWBD denotes Switchboard-1 Release 2 and Switchboard-2.

	Data size	Thread number	Memory usage (G)	Execution time
DNN training (CPU)	600 hours	5	20	60 hours
i-vector extractor training (CPU)	3900 hours	10	100	100 hours
JB training (CPU)	36612 i-vectors	1	4	30 minutes
Ladder network training (GPU)	109667 i-vectors	1	2	3 hours
Processing trials from SAD to i-vector extraction (CPU)	10496 i-vectors	10	70	6 hours
Processing trials after i-vector extraction (CPU)	1986728 trials	1	2	10 minutes

Table 2. The timing report. See Section 3 for detailed explanation.

In addition to using the re-construction errors, the original ladder networks only consider the categorical errors, which are suitable for closed-set identification. For verification (open-set inherently), the features need to be not only separable but also discriminative, so that both compact intra-class variations and separable inter-class differences can be achieved. The categorical loss objective only encourages the separability of features that are not sufficient for verification. Therefore we further add center loss [6] to reduce intra-class variations to learn more discriminative features.

2.3. System 3: System combination

The scores of System 3 are the combination of LLRs from System 1 and System 2. The final scores are calculated as

$$Score = \alpha \cdot LLR_{jb} + (1 - \alpha) \cdot LLR_{ladder} \qquad (3)$$

where α is the interpolation coefficient which is tuned using LDC2016E46-SRE16-CallMyNet-Training-Data labeled part.

3. TIMING REPORT

In Table 2, we show the running time statistics for main modules in our systems. We differentiate the use of CPUs or G-PUs.

In Table 2, processing trials from SAD to i-vector extraction consists of performing SAD, extracting features, extracting statistics and extracting i-vectors. According to Table 2, the single-threaded CPU time and memory used to process an audio file (whether an enrollment file or a test file) are about 6 hours * 10 / 10496 = 20 seconds and 7 G respectively.

Processing trials after i-vector extraction consists of ladder-network based feature transformation (System 2) and JB scoring. According to Table 2, the single-threaded CPU time and memory used to process a trial after obtaining i-vectors for the enrollment file and the test file, are about 10 minutes / 1986728 = 0.3 ms and 2 G respectively.

	System 1(secondary)	System 2	System 3(primary)
	dev		
Equalized			
eer	20.29	19.13	19.18
min_Cprimary	0.8243	0.8225	0.8080
act_Cprimary	0.9095	0.9992	0.9986
Un-equalized			
eer	19.41	19.78	19.55
min_Cprimary	0.8170	0.8213	0.8045
act_Cprimary	0.9209	0.9994	0.9989
	eval		
Equalized			
eer	15.39	17.24	14.71
min_Cprimary	0.7826	0.8570	0.7747
act_Cprimary	0.8993	0.9154	0.8860
Un-equalized			
eer	15.19	17.90	14.58
min_Cprimary	0.8025	0.8795	0.7949
act_Cprimary	0.9205	0.9474	0.9089

Table 3. The performance of the three systems on SRE 2016 ”dev” and ”eval” set. The only difference from our submission result is that here we applied TNorm to system scores, which significantly improve the performance.

4. REFERENCES

[1] “https://github.com/kaldi-asr/kaldi,” .

[2] Dong Chen, Xudong Cao, David Wipf, Fang Wen, and Jian Sun, “An efficient joint formulation for bayesian face verification,” *IEEE Transactions on pattern analysis and machine intelligence*, 2016.

[3] Yiyan Wang, Haotian Xu, and Zhijian Ou, “Joint bayesian gaussian discriminant analysis for speaker verification,” *submit to ICASSP 2016*.

[4] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko, “Semi-supervised learning with ladder networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3546–3554.

[5] Harri Valpola, “From neural pca to deep unsupervised learning,” *Adv. in Independent Component Analysis and Learning Machines*, pp. 143–171, 2015.

[6] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, “A discriminative feature learning approach for deep face recognition,” in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.