Interspeech 2018
2-6 September 2018, Hyderabad

Latent Factor Analysis of Deep Bottleneck Features for Speaker Verification with Random Digit Strings

Ziqiang Shi, Huibin Lin, Liu Liu, Rujie Liu

Fujitsu Research and Development Center

shiziqiang@cn.fujitsu.com

Abstract

Speaker verification with prompted random digit strings has been a challenging task due to very short test utterance. This work investigates how to combine methods from deep bottleneck features (DBF) and latent factor analysis (LFA) to result in a new state-of-the-art approach for such task. In order to provide a wider temporal context, a stacked DBF is extracted to replace the traditional MFCC feature in the derivation of the supervector representations and leads to a significant improvement for the speaker verification. The LFA is used to model these stacked DBFs in both digit and utterance scales. Based on this learned LFA model, two kinds of supervector representations are extracted for utterance and local digits respectively. Since the strengths of DBF and LFA appear complementary, the combination significantly outperforms either of its components. Experiments have been conducted on the public RSR2015 part III data corpus, the results showed that our approach can achieve 1.40% EER and 1.55% EER on male and female respectively.

Index Terms: latent factor analysis, speaker verification, bottleneck feature

1. Introduction

As opposed to text-independent speaker verification, where the speech content is unconstrained, text-dependent speaker verification systems are more favorable for security applications since they showed higher accuracy on short-duration sessions [1, 2]. Text dependent speaker verification has wide applications in many areas, including smart human-machine interface, security, forensic, telephone banking, and so on.

Typical text-dependent speaker verification uses fixed phrase for each user and hence, enrollment and test phrases are matched. For this scenario it is possible that utterance from a user can be recorded beforehand by an imposter and then play it back. This spoofing or attack can be avoided to a certain extent by sharing the same phonetic content but with different context between training and test utterances, for example the user is prompted to utter a digit strings randomly chosen by the system. In this anti-spoofing scenario, the speaker is usually prompted to utter all of 10 digits several times during enrollment and test utterances contain a subset of the digits. This work is tested on part III of the RSR2015 database [1] which is designed to evaluate the ability of a system to deal with this kind of scenario.

Previous methods regarding speaker verification with random prompt digit strings can be grouped into two categories. The first category is based on the traditional state-of-the-art Gaussian mixture model represented universal background model (GMM-UBM) and joint factor analysis (JFA) approach: Larcher et al. [1] use a Hidden Markov Model (HMM) system termed HiLAM to model each speaker and each state corresponding one of the 10 digits; Stafylakis et al. [3] propose to use JFA to extract the global utterance vector and the local digit vector, which are fed into a joint density backend.

In the second category, deep models are ported to speaker verification: deep neural network (DNN) is used to estimate the frame posterior probabilities [4]; DNN as a feature extractor for the utterance level representation [5]; Matejka et al. [6] have shown that using bottleneck DNN features (BN) concatenated to other acoustic features outperformed the DNN method for text-dependent speaker verification; end-to-end deep learning jointly optimizes the speaker representations and models [2]; multi-task deep learning jointly learns both speaker identity and text information [7].

This paper is based on the work of Lee et al. [8], in which deep bottleneck features showed a significant advantage over traditional Mel-frequency cepstral coefficients (MFCC) or shifted-delta-cepstral (SDC) features in language recognition and the work of Stafylakis et al. [3] in which the state-of-the-art JFA approach [9] is employed to extract features which are fed into a joint density backend (JDB) to estimate the log likelihood ratios. We extend and combine these two effective approaches to a new state-of-the-art method for speaker verification with random prompted digit string.

Our contribution is two-fold. Firstly we use the stacked deep bottleneck feature (stack DBF) extracted by using a DNN to replace the traditional MFCC feature to extract the supervector representations [3], which leads to a significant improvement for the speaker verification performance in RSR2015 part III. Secondly multi-scale features including both utterance and digit level supervector representations are employed as frontend and score level calibration and FoCal fusion [10] were further employed to combine subsystem scores into final result.

The remainder of this paper is organized as follows: Section 2 describes the DBF-LFA/JDB approach; The detail experimental results and comparisons are presented in Section 3 and the whole work is summarized in Section 4.

2. Model description

In this section, we describe each component of the approach including stacked deep bottleneck feature, latent factor analysis of stacked DBF and joint density backend of global and local representations.

2.1. Stacked deep bottleneck feature

Performance of speaker verification systems are typically improved by developing robust features which are able to capture relevant speaker characteristics while suppressing channel and session noise. One such feature is the deep

10.21437/Interspeech.2018-1422

bottleneck feature which is widely used in many speech signal processing tasks.

Researchers have proposed to train a DNN in which one of the hidden layers has a small number of units (i.e. the bottleneck layer) to classify senones [11, 12, 13]. In this work, first a speaker-independent GMM-HMM system is trained to classify the utterance frames into senones based on Librispeech corpus [14]. The input feature is 39-dimensional Mel-frequency cepstral coefficients (MFCC, 13 static including the log energy + 13 Δ + 13 $\Delta\Delta$) which are extracted and normalized using utterance-level mean and variance normalization. Then the frame-senon pairs aligned by the GMM-HMM system will be used to train a fully connected DNN. The DNN has 6 hidden layers (with sigmoid activation function) of 2048 nodes each except that the second to last hidden layer has only 64 units. The output layer, which is the classification layer, is a softmax of dimension 9020 i.e., the output layer computes posteriors for 9020 triphone tied states (senones). Once training is complete, all the layers after the bottleneck layer are removed, and the rest of the neural network is used to extract low-dimensional representation of the input. That is each frame of an utterance is forward propagated through the network, and the output activations of all the frames are the so-called DBFs.

Once the first DNN is trained and fixed, we can feed the DBF as inputs to a second DNN giving rise to the stacked DBF [15, 16]. Our stacked DBFs cover a temporal context of 5 frames in the first DNN and 10 frames for the second DNN. After we finished training the second DNN, the two DNNs are stacked together to generate the stacked DBFs. Figure 1 shows the framework of the stacked bottleneck DNN training.

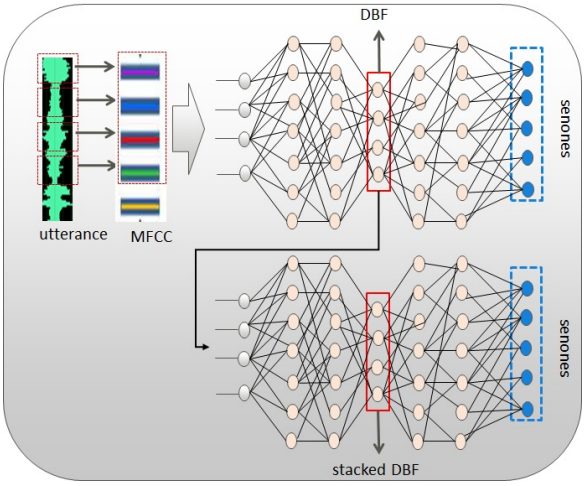

Figure 1: *Illustration of the stacked DBF neural network training.*

2.2. Latent factor analysis of stacked DBF

Latent factor analysis (LFA) is first proposed as decomposed method to transform a speaker model into three different components: a speaker-session-independent component, a speaker dependent component and a session dependent component [17]. Indeed it is a simplification of the popular and effective joint factor analysis (JFA) [9]. Original JFA or LFA is used as a monolithic classifier. In this work they are employed as feature extractors and it is believed that when as a feature extractor LFA is more efficient and effective than JFA for short utterance (<3s). Since both the local and global low rank vectors [3] show poor performance in our empirical study, only the LFA model involving original supervector speaker representation is used as the frontend.

In this work, the speaker model is define as the concatenation of the GMM component means. Let D be the dimension of the feature space, that is the dimension of the stacked DBF. The dimension of a supervector mean is MD where M is the number of Gaussian in the GMM. Assume that the training data consists of I speakers each with H_i sessions, LFA models data generation using the following equation:

$$m_{i,h} = m + \mathbf{D}z_i + \mathbf{U}x_{i,h} \tag{1}$$

where $m_{i,h}$ is the session-speaker dependent supervector mean, $\mathbf{D}$ is $MD \times MD$ diagonal matrix, z_i the speaker vector (a MD vector), $\mathbf{U}$ is the session variability matrix of low rank R and $x_{i,h}$ are the channel factors, a R vector. Both z_i and $x_{i,h}$ are normally distributed among $N(0, I)$.

Once the training is complete, the LFA model (1) is used to compute the speaker vector z_i, which will be used as a representation of the very short utterance, and this z_i will be used in the speaker verification. This speaker vector z_i is in the level of utterance, however in the task of speaker verification with random prompted digit strings, since different speakers have their own characterizations in pronouncing each same digit. This digit-dependent characterization will definitely help in this task. In order to extract digit-dependent z_i, we trained a DNN-HMM automatic speech recognition (ASR) system based on the Librispeech corpus [14] to do segmentation of the utterances in to digits (more specifically to do the alignment between the utterance and the prompt digit string) and then a digit LFA

$$m_{i,d,h} = m + \mathbf{D}z_{i,d} + \mathbf{U}x_{i,h} \tag{2}$$

is constructed, where $m_{i,d,h}$ is the session-speaker-digit dependent supervector mean, $z_{i,d}$ the speaker-digit vector and $x_{i,h}$ are the channel factors, which are digit-independent. Both $z_{i,d}$ and $x_{i,h}$ are normally distributed among $N(0, I)$.

2.3. Joint density backend

Assume z_i and $z_{i,d}$ are all extracted for all enrollment and test utterances. Joint density backend (JDB) believes that $P(z_t, z_s|\text{same-speaker})$ and $P(z_t, z_s|\text{different-speakers})$ (or $P(z_{t,d}, z_{s,d}|\text{same-speaker})$ and $P(z_{t,d}, z_{s,d}|\text{different-speakers})$) both follow normal distributions where given a test vector z_t (or $z_{t,d}$) and an enrolled model z_s (or $z_{s,d}$) [3].

Follow the strategy of Stafylakis et al. [3], first we estimate the $P(z_t, z_s|\text{same-speaker})$ by concatenating z_s and z_t into pairs where enrollment and test vectors belong to the same speaker and estimate the mean μ_0 and covariance matrix Σ_0 of these concatenated vectors. That is

$$\mu_0 = \mathrm{E}\left[\begin{bmatrix} z_s \\ z_t \end{bmatrix}\right] \tag{3}$$

and

$$\Sigma_0 = \mathrm{E}\left[\begin{bmatrix} z_s \\ z_t \end{bmatrix}[z_s^T\, z_t^T]\right] - \mu\mu^T = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}. \tag{4}$$

For the distribution $P(z_t, z_s|\text{different-speakers})$ of $\begin{bmatrix} z_s \\ z_t \end{bmatrix}$ that comes from different speakers, it is assumed that has the same

mean μ_0 and while the covariances matrix is obtained by setting the entries of the off-diagonal blocks A and C equal to zero.

In our practice, empirical study shows there will be about a 20% absolute increase in equal error rate (EER) if full matrix is used for A, B and C. Thus in order to obtain a good result we should make the covariance matrix A, B and C diagonal, which can be easily achieved by estimated the covariance matrix Σ for z_s and z_t separately for each component corresponding to the original GMM.

After the training of the JDB model we treat the verification as a kind of hypothesis testing problem with the null hypothesis $\mathcal{H}_0$ where both z_s and z_t belong to the same speaker and the alternative hypothesis $\mathcal{H}_1$ where they do not. Then the log likelihood ratio score is

$$l(z_t, z_s) = \log \frac{P(z_t, z_s|\mathcal{H}_0)}{P(z_t, z_s|\mathcal{H}_1)} = \log \frac{\mathcal{N}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} |\mu_0, \Sigma_0)}{\mathcal{N}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} |\mu_1, \Sigma_1)}$$

$$= -\frac{1}{2}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} - \mu_0)\Sigma_0^{-1}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} - \mu_0)$$

$$+\frac{1}{2}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} - \mu_1)\Sigma_1^{-1}(\begin{bmatrix} z_t \\ z_s \end{bmatrix} - \mu_1) + \text{const.}$$

It is almost the same process to estimate the JDB model and do verification based on learned JDB for local digit-dependent representations $z_{t,d}$ and $z_{s,d}$. Note that since on RSR2015 part III each test utterance contains 5 digits, in order to do the verification based on speaker-digit vectors $z_{i,d}$ 5 digits log likelihood ratio scores are computed and averaged to be the final score.

2.4. Score normalization

In order to transform log likelihood ratio scores from different speakers into a similar range by using

$$s' = \frac{s - \mu_I}{\sigma_I}$$

so that a common threshold can be used, where μ_I and σ_I are the approximated mean and standard deviation of the impostor score distribution respectively. We tried three score normalization method: zero normalization (z-norm) uses a batch of non-target utterances against the target model to compute the mean μ_I and standard deviation σ_I; test normalization (t-norm) uses the unknown speaker's feature vectors against a set of impostor models to compute the statistics; the zero and test normalized scores are finally averaged to form the s-normalized scores [3].

It is worth noting that because the digit string in RSR 2015 part III is not truly random, the test recording used in the application contains 60 digit strings. Therefore, the impostor recordings used in z-norm contains only the above-mentioned digit strings.

2.5. Score calibration and fusion

Assume there are N subsystems, scores from all subsystems were combined with a linear weighted fusion as follows

$$s = \sum_{n=1}^{N} w_n s_n + b$$

where $\{w_n\}_{n=1}^N$ and b are the weights and the bias. The fusion parameters are trained on the RSR2015 part III development set optimizing the cost C_{llr} [10] function assuming a binary-class regression model in forming the class posterior. The above calibration and fusion can be carried out easily using the FoCal toolkit [10].

3. Experiments

In this section, we describe the experimental setup and results for the proposed method on the public RSR2015 part III English corpus [1].

3.1. Experimental setup

RSR2015 corpus [1] was released by I2R, and it is used to evaluate the performance of different speaker verification systems. In this work, we follow the setup of [3], the part III of RSR2015 is used for the testing of our method. Part III of RSR2015 contains 300 speakers speaking in English and chosen so that they form a representative sample of the Singaporean population. All speech files are of 16kHz. The gender distribution is balanced on the data set (157 male and 143 female). Six mobile devices were used for the recordings that took place under a typical office environment. The speakers are divided into three disjoint groups refereed to as background, development and evaluation, of 97, 97 and 106 speakers respectively. Each speaker model is enrolled with 3 10-digit utterances, recorded with the same handset, while each speaker contributes 3 different speaker models. Each test utterance contains a quasi-random string of 5 digits, one out of 52 unique strings. For both types of utterances, the digit string is given and the verification algorithm may use it. In Table I, the number of trials used for the experiments are given for each set and gender[1].

Table 1: *Trial statistics for RSR2015 digits per set and gender.*

Set	Gender	#target	#nontarget
Dev	Male	5154	251310
Dev	Female	5052	231155
Eval	Male	5943	332863
Eval	Female	5283	253584

3.2. Results and discussion

Eight systems are evaluated and compared across above conditions:

- **MFCC-GMM-UBM**: the standard MFCC with GMM-UBM system.
- **SDBF-GMM-UBM**: the stacked DBF with GMM-UBM system.
- **SDBF-DIGIT-GMM-UBM**: the stacked DBF with digit level GMM-UBM system.
- **SDBF-UTTZ**: the utterance level z_i supervector representation extracted using the stacked DBFs with cosine similarity.
- **SDBF-DIGITZ**: the digit level $z_{i,d}$ supervector representation extracted using the stacked DBFs with average cosine similarity across the utterance.

[1]The numbers of trials are the same as the work [1] and a little different from [3] of Dr. Stafylakis, since they rejected some utterances due to duration and SNR constrains.

- **SDBF-UTTZ-JDB**: the utterance level z_i supervector representation extracted using the stacked DBFs with log likelihood scores of JDB.
- **SDBF-DIGITZ-JDB**: the digit level $z_{i,d}$ supervector representation extracted using the stacked DBFs with average log likelihoods scores of JDB across the utterance.
- **Fusion**: fusion of all the seven systems based on the stacked DBFs, including SDBF-GMM-UBM, SDBF-DIGIT-GMM-UBM, SDBF-UTTZ, SDBF-DIGITZ, SDBF-UTTZ-JDB, and SDBF-DIGITZ-JDB.

Table 2 and Table 3 compare the performances of all above-mentioned systems in terms of equal error rate (EER) on the development and evaluation sets of RSR2015 part III respectively. Obviously stacked DBF is superior to the standard MFCC feature in this task, regardless of the test database and the backend used, when compared with results in [3].

The experimental results show that after using z-norm, EER decreased but the amplitude was significantly different. Among them, the improvement of JDB based systems was the most obvious and the average reduction was about 0.8%. The EER on the female development data decreased the most and nearly about 1.3%. However, the scores for GMM-UBM and cosine distance are not significant, and the EER is reduced by only about 0.3%. The same rule exists for s-norm, but the reduction in EER is reduced. The highest EER reduction for JDB based system is 0.36%. Overall, z-norm and s-norm have a significant improvement on female data sets. It can be seen from the results that the fusion can obtain the state-of-the-art performance.

In addition, from the analysis of the vocabulary table of RSR2015 part III, it can be seen that the probability of occurrence of each digit of all the test recordings for a speaker's one channel is uniform, but due to the error of ASR, individual digital recording clips are mistakenly rejected, eventually leading to each bit. The probability of appearance of digits is not uniform, and the digital probability of each speaker becomes less predictable. However, the output score of the JDBs on the digits is dependent on the number, which directly leads to the use of the conventional z-norm score is not reasonable.

Table 2: *Performance of different systems on the development set of RSR2015 part III in terms of equal error rate (EER %) for male/female.*

EER(%)	w/o norm	z-norm	t-norm	s-norm
M-G-U[1]	5.05/6.34	5.01/6.58	4.65/6.42	4.69/6.50
S-G-U	4.50/5.93	4.32/5.83	3.91/5.15	3.86/5.21
S-D-G-U	4.18/5.35	3.91/4.85	3.50/4.63	3.42/4.53
S-U	3.09/3.57	2.88/3.31	2.69/3.13	2.65/3.06
S-D	4.13/4.20	3.83/3.82	3.27/3.70	3.21/3.51
S-U-J	3.41/3.80	3.17/2.96	2.75/3.56	2.46/2.63
S-D-J	4.57/5.47	4.14/3.38	3.65/3.96	3.31/3.24

4. Conclusions

In this paper we investigated the effectiveness of latent factor analysis (LFA) modeling for stacked deep bottleneck features (DBF) on the task of speaker verification with random prompt digit strings. Benefits from the strength of deep architecture in modeling data correlation without the need of handcrafted transformation, stacked DBF leads to %10 relative improvement. Further with the help of LFA on both utterance and digit scales to get rid of the channel and session noise we achieve the new state-of-the-art on RSR2015 part III task.

[1]Here M-G-U stands for MFCC-GMM-UBM, and other notations have the same meaning.

Table 3: *Performance of different systems on the evaluation set of RSR2015 part III in terms of equal error rate (EER %) for male/female.*

EER(%)	w/o norm	z-norm	t-norm	s-norm
M-G-U	3.93/6.37	3.81/4.39	3.38/3.66	3.54/3.23
S-G-U	3.54/5.61	3.20/3.42	2.72/2.85	2.70/2.68
S-D-G-U	3.48/3.46	2.98/3.53	2.82/3.34	2.61/3.18
S-U	2.53/2.55	2.28/2.30	1.52/1.84	1.83/1.90
S-D	4.35/3.47	2.96/2.75	2.43/2.90	2.45/2.64
S-U-J	2.19/3.07	2.10/2.38	1.83/2.32	1.57/1.84
S-D-J	3.51/4.55	2.90/2.98	2.68/3.19	2.48/2.56
Fusion	2.13/2.41	1.69/1.72	1.47/1.51	1.40/1.55

5. References

[1] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5115–5119.

[3] T. Stafylakis, M. J. Alam, and P. Kenny, "Text-dependent speaker recognition with random digit strings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1194–1203, 2016.

[4] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[5] E. Variani, X. Lei, E. Mcdermott, and I. L. Moreno, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP 2014 - 2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4052–4056.

[6] H. Zeinali, H. Sameti, L. Burget, J. Cernocky, N. Maghsoodi, and P. Matejka, "i-vector/hmm based text-dependent speaker verification system for reddots challenge," in *INTERSPEECH*, 2016.

[7] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verificaion," in *INTERSPEECH*, 2015.

[8] K. A. Lee, H. Li, L. Deng, V. Hautamaki, W. Rao, X. Xiao, A. Larcher, H. Sun, T. H. Nguyen, G. Wang, A. Sizov, J. Chen, I. Kukanov, A. H. Poorjam, T. N. Trong, C.-L. Xu, H. Xu, B. Ma, E. S. Chng, and S. Meignier, "The 2015 nist language recognition evaluation: The shared view of i2r, fantastic4 and singams." in *Interspeech 2016*, vol. 2016, 2016, pp. 3211–3215.

[9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[10] N. Brummer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scorestutorial and user manual," *Software available at http://sites. google. com/site/nikobrummer/focalmulticlass*, 2007.

[11] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *INTERSPEECH*, 2011, pp. 237–240.

[12] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4153–4156.

[13] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLOS ONE*, vol. 9, no. 7, 2014.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[15] H. Xu, H. Su, C. E. Siong, and H. Li, "Semi-supervised training for bottle-neck feature based dnn-hmm hybrid systems," in *INTERSPEECH*, 2014, pp. 2078–2082.

[16] H. Xu, V. H. Do, X. Xiao, and E. Chng, "A comparative study of bnf and dnn multilingual training on cross-lingual low-resource speech recognition." in *INTERSPEECH*, 2015, pp. 2132–2136.

[17] J.-F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. W. D. Evans, B. G. B. Fauve, and J. S. D. Mason, "Alize/spkdet: a state-of-the-art open source software for speaker recognition," *Odyssey 2008: The Speaker and Language Recognition Workshop*, p. 20, 2008.