Interspeech 2018
2-6 September 2018, Hyderabad

Joint Learning of J-Vector Extractor and Joint Bayesian Model for Text Dependent Speaker Verification

Ziqiang Shi, Liu Liu, Huibin Lin, Rujie Liu

Fujitsu Research and Development Center

shiziqiang@cn.fujitsu.com

Abstract

J-vector and joint Bayesian have been proved to be very effective in text dependent speaker verification with short-duration speech. However current state-of-the-art framework often consider training the J-vector extractor and the joint Bayesian classifier separately. Such an approach will result in information loss for j-vector learning and also fail to exploit an end-to-end framework. In this paper we present a integrated approach to text dependent speaker verification, which consists of a siamese deep neural network that takes two variable length speech segments and maps them to the likelihood score and speaker/phrase labels, where the likelihood score as a loss guide is computed by a variant joint Bayesian model. The likelihood loss guide can constrain the j-vector extractor for improving the verification performance. Since the strengths of j-vector and joint Bayesian analysis appear complementary the joint learning significantly outperforms traditional separate training scheme. Our experiments on the the public RSR2015 part I data corpus demonstrate that this new training scheme can produce more discriminative j-vectors and leading to performance improvement on the speaker verification task.

Index Terms: speaker verification, siamese network, joint learning, j-vector, joint Bayesian analysis

1. Introduction

Text-dependent speaker verification has lexical constraints which require the matching of both voice characteristics and the pass-phrases being spoken. As opposed to text-independent speaker verification, where the speech content is unconstrained, text-dependent systems are much preferred for security applications since they showed higher accuracy on short-duration sessions [1, 2].

In the literature, the previous methods considered for text-dependent speaker verification can be grouped into two categories. The first category is based on the traditional state-of-the-art GMM-UBM or i-vector approach, which may not work well in this case [3, 1, 4]. Larcher et al. [1] use a Hidden Markov Model (HMM) system termed HiLAM to model each speaker and each state corresponding senones; Stafylakis et al. [5] propose to use JFA to extract global utterance vector and local 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 [6]; DNN as a feature extractor for the utterance level representation [7]; Matejka et al. [8] have shown that using bottle-neck 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 [9].

This paper is based on the work of Chen et al. [9], in which the j-vector was introduced as a kind of more compact representation for text dependent utterances, and the classic probability linear discriminant analysis (PLDA) was used as the back-end classifier [10, 11] and work of [12], in which the state-of-the-art joint Bayesian approach is proposed to model the two facial images jointly with an appropriate prior that considers intra- and extra-personal variations over the image pairs. Since the feature extraction and classification are completely separated, these approaches may lead to information loss and also fail to exploit an end-to-end framework.

In order to develop an integrated framework of j-vector extraction and joint Bayesian modeling for text-dependent speaker verification, we propose to add the likelihood score of a variant joint Bayesian as a loss guide to constrain the j-vector extractor network for improving the verification performance. Specifically we construct a siamese network has two same branches with shared weights, where each branch is a usual j-vector extractor and has its own multi-task cross-entropy loss. The proposed likelihood loss guide is added to the output layer of the siamese network as a constraint of the j-vector extraction. This **j**oint learning of **j**-vector extractor network and **j**oint Bayesian model will be called J3 in this work.

Our contribution is two-fold. Firstly we integrate the joint Bayesian into the training of j-vector extraction network, make two steps of speaker verification into a novel unified J3 deep learning architecture, which leads to a significant improvement for the speaker verification performance on RSR2015 part III. Secondly in order to obtain better j-vector and verification performance through J3, we propose a novel training scheme that keep a snapshot of the joint Bayesian which is updated once in m ($m \geq 1$) epochs and fixed such snapshot during the updating of the j-vector extraction network in these m epochs.

The remainder of this paper is organized as follows: Section 2 reviews the standard j-vector and joint Bayesian approach. Section 3 describes the approach of joint learning of j-vector extractor and joint Bayesian model for text dependent speaker verification. The detail experimental results and comparisons are presented in Section 4 and the whole work is summarized in Section 5.

2. J-vector and joint Bayesian baseline approach

The standard j-vector system introduced in [9] and joint Bayesian model [12] is used as the baseline in this work. This section gives a brief review of this baseline.

2.1. J-vector extraction

Chen et al. [9] proposed a method to train a DNN to make classifications for both speaker and phrase identities by

10.21437/Interspeech.2018-1500

minimizing a total loss function consisting a sum of two cross-entropy losses as shown in Fig. 1 - one related to the speaker label and the other to the text label. Once training is complete, the output layer is removed, and the rest of the neural network is used to extract speaker-phrase joint features. That is each frame of an utterance is forward propagated through the network, and the output activations of all the frames are averaged to form an utterance-level feature called j-vector. The enrollment speaker models are formed by averaging the j-vectors corresponding to the enrollment recordings.

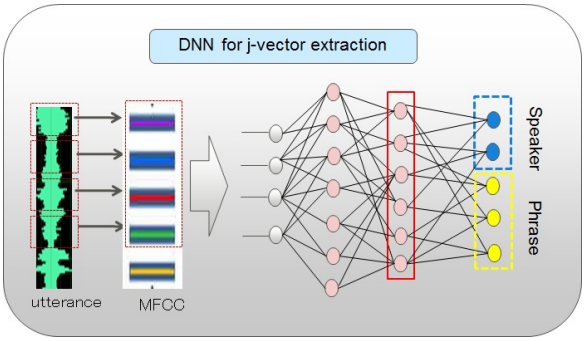

Figure 1: *Multi-task joint learning DNN as j-vector extractor.*

2.2. The joint Bayesian model

For the back-end, the state-of-the-art joint Bayesian model [12] is employed as a classifier for speaker verification. Classical joint Bayesian model assumes the observed feature, e.g. the j-vector, as the result of a generative model. For simplicity of notation joint Bayesian model with only single speaker label is used as an example. Assume that the training data consists of I speakers each with H_i sessions, joint Bayesian models data generation using the following equation:

$$x_{ij} = \mu + z_i + \epsilon_{ij}.$$

z_i and ϵ_{ij} are defined to be Gaussian with diagonal covariance Σ_z and Σ_ϵ respectively. Let $\theta = \{\mu, \Sigma_z, \Sigma_\epsilon\}$, $x_i = \{x_{ij} : j = 1, ..., H_i\}$, $X = \{x_{ij} \in \mathbb{R}^D : i = 1, ..., I; j = 1, ..., H_i\}$, and the term μ represents the overall mean of the training vectors.. More formally the model can be described in terms of conditional probabilities:

$$\begin{aligned} p(x_{ij}|z_i, \theta) &= \mathcal{N}(x_{ij}|\mu + z_i, \Sigma_\epsilon), \\ p(z_i) &= \mathcal{N}(z_i|0, \Sigma_z), \end{aligned}$$

where $\mathcal{N}(x|\mu, \Sigma)$ represents a Gaussian in x with mean μ and covariance Σ.

The parameters θ of this joint Bayesian model can be estimated using the Expectation Maximization (EM) [13] algorithm. The auxiliary function for EM is

$$\begin{aligned} &Q(\theta|\theta_t) = \mathrm{E}_{U,V|X,\theta_t}[\log p(X, U, V|\theta)] \\ =&\mathrm{E}_{U,V|X,\theta_t}\left\{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{H_{ij}} \log[p(x_{ijk}|u_i, v_j, \theta)p(u_i, v_j)]\right\} \end{aligned}$$

E steps: calculate the expectation $\mathrm{E}_{Z|X,\theta_t}[z_i]$ and $\mathrm{E}_{Z|X,\theta_t}[z_i z_i^T]$. We have

$$\mathrm{E}_{Z|X,\theta_t}[z_i] = \left(\Sigma_z^{-1} + H_i\Sigma_\epsilon^{-1}\right)^{-1}\Sigma_\epsilon^{-1}\sum_{j=1}^{H_i}(x_{ij} - \mu)$$

and

$$\mathrm{E}_{Z|X,\theta_t}[z_i z_i^T] = \left(\Sigma_z^{-1} + H_i\Sigma_\epsilon^{-1}\right)^{-1} + \mathrm{E}_{Z|X,\theta_t}[z_i]\mathrm{E}_{Z|X,\theta_t}[z_i]^T.$$

M steps: update the parameter θ. Indeed we have

$$\Sigma_z = \frac{1}{\sum_{i=1}^{I}\sum_{j=1}^{H_i} 1}\sum_{i=1}^{I}\sum_{j=1}^{H_i}\mathrm{E}_{Z|X,\theta_t}[z_i z_i^T],$$

$$\begin{aligned} \Sigma_\epsilon &= \frac{1}{\sum_{i=1}^{I}\sum_{j=1}^{H_i} 1}\sum_{i=1}^{I}\sum_{j=1}^{H_i}\{(x_{ij} - \mu)(x_{ij} - \mu) \\ &- 2(x_{ij} - \mu)\mathrm{E}_{Z|X,\theta_t}[z_i]^T + \mathrm{E}_{Z|X,\theta_t}[z_i z_i^T]\}, \end{aligned}$$

and

$$\mu = \frac{\sum_{i=1}^{I}\sum_{j=1}^{H_i} x_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{H_i} 1}.$$

With the learned joint Bayesian model, given a test x_t and an enrolled model x_s, the likelihood ratio score is

$$\begin{aligned} l(x_t, x_s) &= \frac{P(x_t, x_s|\text{same-speaker})}{P(x_t, x_s|\text{different-speakers})} \\ &= \frac{\int p(x_t, x_s, z|\theta)dz}{\int p(x_t, z_t|\theta)dz_t \int p(x_s, z_s|\theta)dz_s} \\ &= \frac{\int p(x_t, x_s|z, \theta)p(z)dz}{\int p(x_t|z_t, \theta)p(z_t)dz_t \int p(x_s|z_s, \theta)p(z_s)dz_s} \\ &= \frac{\mathcal{N}(\begin{bmatrix} x_t \\ x_s \end{bmatrix} | \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Sigma_z + \Sigma_\epsilon & \Sigma_z \\ \Sigma_z & \Sigma_z + \Sigma_\epsilon \end{bmatrix})}{\mathcal{N}(x_t|\mu, \Sigma_z + \Sigma_\epsilon)\mathcal{N}(x_s|\mu, \Sigma_z + \Sigma_\epsilon)}. \end{aligned}$$

Whereas the multi-task objective has been shown to enable the training of DNNs for j-vector extraction for challenging short duration text-dependent speaker verification problems, a disadvantage of the baseline with j-vector/joint Bayesian is that the post-classifier is not part of the original objective function. On the other hand for joint Bayesian analysis the objective function minimizing during training is directly related to the speaker/phrase verification accuracy. We seek to combine the benefits of both frontend and backend in a strategy reminiscent of siamese network except that here we still keep the multi-task objective for j-vector extraction. This intuitive idea result in the following J3 architecture.

3. J3 architecture

Generally speaking J3 is a siamese DNN with an additional multi-task loss on each branch. We extent the architecture in Figure 1 in order to created a siamese network as in Figure 2 which we refer as J3 network with one original objective in j-vector extraction network and the other objective is obtained by using joint Bayesian resulted log likelihood of the input two j-vectors as a guide.

3.1. Training of J3

In order to obtain a better performance, we firstly do the initialization of J3 network by pre-train an initial j-vector extraction neural network on the background data of RSR2015 part III by optimizing multiple classification-based cross entropy loss objective functions under the guide of the speaker and phrase ID information. For the speaker verification task, an effective speaker representation is necessary or the first

step to achieve a good performance. These j-vectors learned from the background data is employed for training a good initial joint Bayesian model of j-vector verification, which can generate some useful knowledge to further guide the process of network optimization. Furthermore, the j-vector is also the basic of learning a distance for speaker verification in the whole framework.

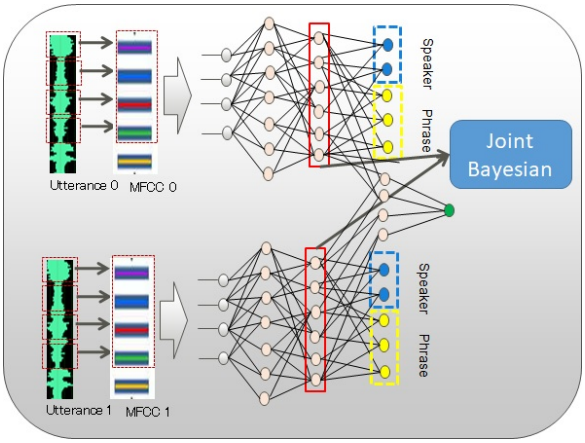

Figure 2: *The proposed J3 framework of text-dependent speaker verification.*

After the initialization (or pre-training) of both the j-vector extractor network and the joint Bayesian model, then we can start to train the J3 network. The training will last for several epochs periodically. More specifically, we maintain a snapshot of the joint Bayesian model which is updated periodically, say after every m ($m \geq 1$) epochs. Whenever the joint Bayesian model is updated, it will be used as a additional log likelihood loss to guide the training of j-vector extraction network.

We explain the idea and process in detail. In each period there are two steps: one step is to update the joint Bayseian model with the j-vector extractor network fixed, that is all utterance pairs are fed into the j-vector extractor network and result in output j-vector pairs (x_t, x_s) which are used in the update or training of the joint Bayesian model (by using the method in Section 2); the other step is to update the j-vector network with fixed joint Bayesian model, where three kinds of supervised information are used to modify the gradient direction, one is the original *m*ulti-*t*ask *c*ross *e*ntropy loss

$$l_{mtce} = -\sum_{t}\sum_{c} 1_{\{l_t=c\}} \log p(x_t, W)$$

in j-vector network training in Section 2, where W the parameter of the J3 network, x_t and l_t are the input and the ground-truth label respectively. The second loss is the traditional training criterion provided by the *sia*mese network

$$l_{sia} = \sum_{t,s} [(1-b)\exp(S(x_t, x_s, W)) + b\exp(-S(x_t, x_s, W))]$$

which is to increase the similarity between true pairs and reduce that of imposter pairs, where $S(x_t, x_s, W)$ is the similarity output produced by the siamese network and b is the true binary label. The third loss is provide by the square difference of score produced by the joint Bayesian model and output of the siamese network

$$l_j = \frac{1}{N}\sum_{t,s}(S(x_t, x_s, W) - J(x_t, x_s))^2$$

where $J(x_t, x_s)$ is the log likelihood score provide by the snapshot of the joint Bayesian model. Finally these three losses l_{mtce}, l_{sia} and l_j are summed as the whole loss, which is used to update the parameter W of the siamese network.

Algorithm 1 Joint training of J3 network

Input: start with random weights of the siamese network and the parameters of joint Bayesian; utterance samples and labels.
0: **initialization**: pre-train the branch of j-vector extractor network in siamese network with utterances as inputs and speaker/phrase IDs as outputs using multi-task learning in Section 2, result in a j-vector extractor which is used to extract j-vectors to pre-train the joint Bayesian classifier.
1: **repeat**: **for** epoch = 1,2,...
2: **if** mod(epoch,m)==0; **then** fix the siamese network, and feed all utterance pairs into the j-vector network and result in output j-vector pairs (x_t, x_s) which are used to update of the joint Bayesian model.
3: **else** fix the joint Bayesian model, update the siamese network with the loss $l_{mtce} + l_{sia} + l_j$ using stochastic gradient method.
4: **until** stopping conditions are satisfied.
Output: J3 network.

3.2. Verification with J3

In the test process, we feed each pair of utterances into our J3 network framework and output a predicted similarity for the speaker verification task. Obviously here are two paths of verification: one is directly use the output of the siamese network that is the $S(x_t, x_s, W)$; the other is to use siamese network as the j-vector extractor and the joint Bayesian as a classifier. Empirical experiments show that both methods outperform traditional separately trained methods

3.3. 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 [5]. Finally since s-norm gets the best performance, we only report the results of s-norm in this work.B

4. Evaluation and discussion

In this section, we describe the experimental setup and results for the proposed method on the public RSR2015 English corpus [1] and our internal Huiting202 Chinese Mandarin database collected by the Huiting Techonogly[2].

[2]http://huitingtech.com/

4.1. Experimental setup

RSR2015 corpus [1] was released by I2R, is used to evaluate the performance of different speaker verification systems. In this work, we follow the setup of [14], the part I of RSR2015 is used for the testing of J3. The background and development data of RSR2015 part I are merged as new background data to train the J3.

Our internal gender balanced Huiting202 database is designed for local applications. It contains 202 speakers reading 20 different phrases, 20 sessions each phrase. All speech files are of 16kHz. 132 randomly selected speakers are used for training the background multi-task learned DNN, and the remaining 70 speakers were used for enrollment and evaluation.

In this work, 39-dimensional Mel-frequency cepstral coefficients (MFCC, 13 static including the log energy + 13 Δ + 13 $\Delta\Delta$) are extracted and normalized using utterance-level mean and variance normalization. The input is stacked normalized MFCCs from 11 frames (5 frames from each side of the current frame). The DNN branch used in J3 has 6 hidden layers (with sigmoid activation function) of 2048 nodes each. The J3 network is trained by using the Algorithm 1. Once the J3 is trained, the j-vector can be extracted during the enrollment and evaluation stages.

4.2. Results and discussion

Four systems are evaluated and compared across above conditions:

- **j-vector**: the standard j-vector system with cosine similarity.
- **joint Bayesian**: the j-vector system with classic joint Bayesian in [12].
- **J2**: joint training of j-vector extractor and joint Bayesian, and use the siamese network as the j-vector extractor and the joint Bayesian as a backend as described in Section 3.
- **J3**: joint training of j-vector extractor and joint Bayesian, and directly use the similarity output of the siamese network to do verification as described in Section 3.

When evaluation a speaker is enrolled with 3 utterances of the same phrase. The task concerns on both the phrase content and speaker identity. Nontarget trials are of three types: the impostor pronouncing wrong lexical content (impostor wrong, IW); a target speaker pronouncing wrong lexical content (target wrong, TW); the imposter pronouncing correct lexical content (impostor correct, IC).

The class defined in **j-vector** in all models is the multi-task label of both the speaker and phrase. The **joint Bayesian** method is trained using the j-vectors, the number of principle components is set to 100 and then the joint Bayesian model is estimated with 10 iterations. For **J2** and **J3**, the setting of the branch DNN is the same as j-vector extractor network in **j-vector**, the setting of joint Bayesian component is the same as **joint Bayesian**.

Table 1 and 2 compare the performances of all above-mentioned systems in terms of equal error rate (EER) for the three types of nontarget trials. Obviously both **J3** and **J2** is superior to the standard **joint Bayesian** and **j-vector**, regardless of the test database. Since joint training system can explore both the j-vector extractor network and joint Bayesian model, and further help and guide each other in training to improve, it constantly performs better than standard systems.

Table 1: *Performance of different systems on the evaluation set of RSR2015 part I in terms of equal error rate (EER %).*

EER(%)	j-vector	joint Bayesian	J2	J3
IW	0.95	0.02	0.02	0.02
TW	3.14	0.03	0.02	0.02
IC	7.86	3.61	2.81	2.42
Total	1.45	0.46	0.35	0.28

Table 2: *Performance of different systems on the evaluation set of Huiting202 in terms of equal error rate (EER %).*

EER(%)	j-vector	joint Bayesian	J2	J3
IW	0.86	0.10	0.08	0.07
TW	6.71	0.04	0.04	0.03
IC	4.57	2.52	2.07	1.87
Total	1.37	0.45	0.28	0.23

5. Conclusion

In this paper we have proposed J3 a joint learning approach to integrated j-vector extractor and the joint Bayesian model into one unified framework. The most important advantages of J3, compared to standard j-vector with joint Bayesian, is that in J3 both j-vector extraction component and joint Bayesian component can help to improve each other during the joint training. The jointly optimized hybrid network outperformed both the plain j-vector and joint Bayesian methods. Reported results showed that J3 provided significant reduction in error rates over conventional systems in term of EER.

6. 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] P. Kenny, T. Stafylakis, P. Ouellet, and M. J. Alam, "Jfa-based front ends for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1705–1709.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] S. Ioffe, "Probabilistic linear discriminant analysis," *Proc ECCV*, vol. 22, no. 4, pp. 531–542, 2006.

[11] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision, 2007. Proceedings*, 2007, pp. 1–8.

[12] D. Chen, X. Cao, D. Wipf, F. Wen, and J. Sun, "An efficient joint formulation for bayesian face verification," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 32–46, 2017.

[13] A. P. Dempster, "Maximum likelihood estimation from incomplete data via the em algorithm (with discussion," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[14] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.