GAUSSIAN-CONSTRAINED TRAINING FOR SPEAKER VERIFICATION

Lantian Li, Zhiyuan Tang, Ying Shi, Dong Wang

Center for Speech and Language Technologies, RIIT, Tsinghua University, China
Beijing National Research Center for Information Science and Technology

ABSTRACT

Neural models, in particular the d-vector and x-vector architectures, have produced state-of-the-art performance on many speaker verification tasks. However, two potential problems of these neural models deserve more investigation. Firstly, both models suffer from 'information leak', which means that some parameters participating in model training will be discarded during inference, i.e, the layers that are used as the classifier. Secondly, these models do not regulate the distribution of the derived speaker vectors. This 'unconstrained distribution' may degrade the performance of the subsequent scoring component, e.g., PLDA. This paper proposes a Gaussian-constrained training approach that (1) discards the parametric classifier, and (2) enforces the distribution of the derived speaker vectors to be Gaussian. Our experiments on the *VoxCeleb* and *SITW* databases demonstrated that this new training approach produced more representative and regular speaker embeddings, leading to consistent performance improvement.

Index Terms— speaker verification, deep neural network

1. INTRODUCTION

Automatic speaker verification (ASV) is an important biometric authentication technology and has found a broad range of applications. The current ASV methods can be categorized into two groups: the statistical model approach that has gained the most popularity [1, 2, 3], and the neural model approach that emerged recently but has shown great potential [4, 5, 6].

Perhaps the most famous statistical model is the Gaussian mixture model−universal background model (GMM-UBM) [1]. It factorizes the variance of speech signals by the UBM, and then models individual speakers conditioned on that factorization. More succinct models design subspace structures to improve the statistical strength, including the joint factor analysis model [2] and the i-vector model [3]. Further improvements were obtained by either discriminative

models (e.g., PLDA [7]) or phonetic knowledge transfer (e.g., the DNN-based i-vector model [8, 9]).

The neural model approach has also been studied for many years as well, however it was not as popular as the statistical model approach until recently training large-scale neural models became feasible. The initial success was reported by Ehsan et al. on a text-dependent task [4], where frame-level speaker features were extracted from the last hidden layer of a deep neural network (DNN), and utterance-based speaker vectors ('d-vectors') were derived by averaging the frame-level features. Learning frame-level speaker features offers many advantages, which paves the way to deeper understanding of speech signals.

Researchers followed Ehsan's work in two directions. In the first direction, more speech-friendly DNN architectures were designed, with the goal of learning stronger frame-level speaker features while keeping the simple d-vector architecture unchanged [6]. In the second direction, researchers pursue end-to-end solutions which produce utterance-level speaker vectors directly [5, 10, 11, 12]. A representative work in this direction is the x-vector architecture proposed by Snyder et al. [12], which produces the utterance-level speaker vectors (x-vectors) from the first- and second-order statistics of the frame-level features.

For both the d-vector and x-vector architectures, however, there are two potential problems. Firstly, the DNN models involve a parametric classifier (i.e., the last affine layer) during model training. This means that part of the knowledge involved in the training data is used to learn a classifier that will be ultimately thrown away during inference, leading to potential 'information leak'. Secondly, these models do not regulate the distribution of the derived speaker vectors, either at the frame-level or at the utterance-level. The uncontrolled distribution will degrade the subsequent scoring component, especially the PLDA model that assumes the speaker vectors are Gaussian [7].

To deal with these two problems, we propose a Gaussian-constrained training approach in this paper. This new training approach will (1) discard the parametric classifier to mitigate information leak, and (2) enforce the distribution of the derived speaker vectors to be Gaussian to meet the requirement of the scoring component. Our experiments on two databases demonstrated that the approach can produce more representa-
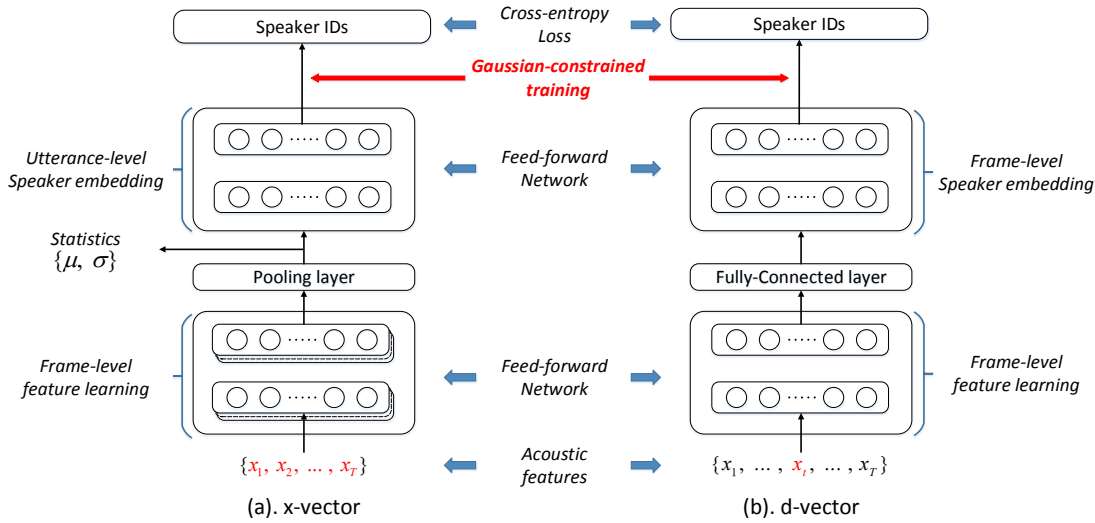
Fig. 1. The architecture of the x-vector and d-vector models, shown in a comparative way.

tive and regular speaker vectors than both the d-vector and x-vector models, which in turn leads to consistent performance improvement.

The organization of this paper is as follows. Section 2 gives a brief overview for the x-vector and d-vector models, and Section 3 presents the proposed Gaussian-constrained training. Experiments are reported in Section 4, and the paper is concluded in Section 5.

2. OVERVIEW OF X-VECTOR AND D-VECTOR MODELS

The x-vector model and the d-vector model are two typical neural models adopted by ASV researchers. The architectures of these models are shown in Figure 1, in a comparative way.

For the x-vector model, it consists of three components. The first component is used for frame-level feature learning. The input of this component is a sequence of acoustic features, e.g., filterbank coefficients. After several feed-forward neural networks, it outputs a sequence of frame-level speaker features. The second component is a statistic pooling layer, in which statistics of the frame-level features, e.g., mean and standard deviation, are computed. This statistic pooling maps a variable-length input to a fixed-dimensional vector. The third component is used to produce utterance-level speaker vectors. The output layer of this component is a softmax, in which each output node corresponds to a particular speaker in the training data. The network is trained to discriminate all the speakers in the training data, conditioned on the input utterance. Once the model is well trained, utterance-level speaker vectors, i.e., x-vectors, are read from a layer in the last component, and a scoring model, such as PLDA, will be used to score the trials.

The d-vector model has a similar but simpler architecture. It consists of two components, one is for frame-level feature

learning, the same as the first component of the x-vector model, and the other is for frame-level speaker embedding, the same as the third component of the x-vector model. Since the entire architecture is frame-based, a pooling layer is not required. The network is trained to discriminate speakers in the training data, but conditioned on each frame. Once the training is done, utterance-level speaker vectors, i.e., d-vectors, are derived by averaging the frame-level features. Finally, a scoring model such as PLDA will be used to score the trials.

3. GAUSSIAN-CONSTRAINED TRAINING

As mentioned in Section 1, both the d-vector and the x-vector models suffer from (1) information leak and (2) unconstrained distribution of speaker vectors. We propose a Gaussian-constrained training approach to solve these problems. In brief, this approach introduces a Gaussian prior on the output of the embedding layer, which can be formulated as a regularization term in the training objective. Training with this augmented objective will enforce the parameters of the classification layer more predictable, so more parameter-free. Meanwhile, it will urge the model producing more Gaussian speaker vectors, at either the frame-level (for d-vector) or the utterance-level (for x-vector).

For a clear presentation, we shall use the x-vector model to describe the process, whereas the same argument applies to the d-vector model in a straightforward way. Specifically, if all the utterance-level x-vectors in the training set have been derived, the speaker-level x-vector can be simply obtained by averaging all the utterance-level x-vectors belong to that speaker. This is formally written as:

$$v(s) = \frac{1}{|\mathcal{E}(s)|} \sum_{x \in \mathcal{E}(s)} f(x), \tag{1}$$

where $\mathcal{E}(s)$ is the set of utterances belonging to speaker s; $f(x)$ is the x-vector of utterance x; $v(s)$ is the speaker-level x-vector.

Based on the speaker-level x-vectors $\{v(s)\}$, each speech utterance x can be easily classified as follows:

$$p(s|f(x)) = \frac{e^{f(x) \cdot v(s)}}{\sum_{s'} e^{f(x) \cdot v(s')}}, \qquad (2)$$

which can be regarded as a non-parametric classifier. If we use this non-parametric classifier to replace the parametric classifier (usually the last affine layer) of the x-vector model in Fig. 1, we reach the *full-info training* proposed by Li et al. [13].

The model can be trained with the classical cross entropy (CE) criterion, written by:

$$L' = - \sum_t \log\ p(s_t|f(x_t)) \qquad (3)$$

where x_t and s_t are the t-th speech utterance and the corresponding ground truth label. Note that the gradients on CE will be *fully* propagated to the weights of the feature learning component, as the parameters of the classifier (2) are $\{v(s)\}$, which are dependent on the feature learning component as well. This means all the knowledge of the data will be exploited to learn the feature component, hence amending the information leak problem.

However, we cannot use a 'virtual classifier' parameterized by $\{v(s)\}$ in a practical implementation; instead, we resort to an engineering solution that designs a true parametric classifier, and replaces the parameters by $\{v(s)\}$ regularly. This train-and-replacement scheme works well in many scenarios [13], but may slow the training or cause fluctuation.

A more elegant approach is to keep the classifier parameters, but introduce a regularization term that encourages the parameters approaching to $\{v(s)\}$. For this purpose, the following regularization term is designed:

$$R = \sum_s \sum_{x_t \in \mathcal{E}(s)} ||f(x_t) - \theta_s||_2, \qquad (4)$$

where θ_s represent the parameters in the classifier that are associated with the output node corresponding to speaker s. With this regularization, the training objective is given by:

$$L = L' + \alpha R,$$

where α controls the strength of the regularization. Clearly, if α is sufficiently large, θ_s will converge to the speaker-level x-vector $\{v(s)\}$. Moreover, it is clear that the regularization term R encourages all the utterance-level x-vectors $f(x_t)$ belonging to speaker s to be a Gaussian $N(\theta_s, \mathbf{I})$. Therefore, we name this new training approach as *Gaussian-constrained training*.

The Gaussian-constrained training possesses several advantages. Firstly, it encourages the parameters of the classifier to converge to the speaker-level x-vectors, which equals to removing these parameters gradually. This mitigates the information leak problem, but does not suffer from the instability of the full-info training. For this reason, Gaussian-constrained training can be regarded as a *soft* full-info training. Secondly, it encourages the utterance-level x-vectors to be Gaussian, which is a key requirement for many scoring models, particularly PLDA. Thirdly, this approach is flexible. We can choose α to control the strength of the regularization, or choose other regularization forms to produce speaker vectors in other forms of distributions.

4. EXPERIMENTS

4.1. Data

Three datasets were used in our experiments: VoxCeleb, SITW and CSLT-SITW. VoxCeleb was used for model training, while the other two were used for evaluation. More information about these three datasets is presented below.

VoxCeleb: A large-scale free speaker database collected by University of Oxford, UK [14]. The entire database involves two parts: *VoxCeleb1* and *VoxCeleb2*. Since part of the speakers is shared by VoxCeleb and SITW, a simple data purging was conducted to remove all the data of the shared speakers. The purged dataset involves $1,236,567$ utterances from $7,185$ speakers. This dataset was used to train both the d-vector model and the x-vector model, plus the LDA and PLDA models. Data augmentation was applied, where the MUSAN corpus [15] was used to generate noisy utterances and the room impulse responses (RIRS) corpus [16] was used to generate reverberant utterances.

SITW: A standard database used to test ASV performance in real-world conditions [17]. It was collected from open-source media channels, and consists of speech data covering 299 well-known persons. There are two standard datasets for testing: *Dev. Core* and *Eval. Core*. We used these two sets to conduct the first evaluation in our experiment. Note that the acoustic condition of SITW is similar to that of the training set VoxCeleb, so this evaluation can be regarded as a condition-matched evaluation.

CSLT-SITW: A small dataset collected by CSLT at Tsinghua University. It consists of 11 speakers, each of which records a couple of Chinese digital strings by several mobiles. Each string contains 8 Chinese digits, and the duration is about 2-3 seconds. The scenarios involve laboratory, corridor, street, restaurant, bus and subway. Speakers varied their poses during the recording, and the mobile phones were placed both near and far. There are $6,915$ utterances in total.

4.2. Settings

For a comprehensive comparison, three baseline systems following the Kaldi SITW recipe [18] were built: an i-vector

system, an x-vector system and a d-vector system.

For the i-vector system, the acoustic feature involves 24-dimensional MFCCs plus the log energy, augmented by the first- and second-order derivatives. We also apply cepstral mean normalization (CMN) and the energy-based voice active detection (VAD). The UBM consists of $2,048$ Gaussian components, and the dimensionality of the i-vector space is 400. LDA is applied to reduce the dimensionality of i-vectors to 150 prior to PLDA scoring.

For the x-vector system, the feature-learning component is a 5-layer time-delay neural network (TDNN). The slicing parameters for the five TD layers are: $\{t\text{-}2, t\text{-}1, t, t\text{+}1, t\text{+}2\}$, $\{t\text{-}2, t, t\text{+}2\}$, $\{t\text{-}3, t, t\text{+}3\}$, $\{t\}$, $\{t\}$. The statistic pooling layer computes the mean and standard deviation of the frame-level features from a speech segment. The size of the output layer is $7,185$, corresponding to the number of speakers in the training set. Once trained, the 512-dimensional activations of the penultimate hidden layer are read out as an x-vector. This vector is then reduced to a 150-dimensional vector by LDA, and finally the PLDA model is employed to score the trials. Refer to [19] for more details. In the Gaussian-constrained training, the hyper-parameter α is empirically set to 0.05.

For the d-vector system, the DNN structure is similar to that of the x-vector system. The only difference is that the statistic pooling layer in the x-vector model is replaced by a TD layer whose slicing parameter is set to $\{t\text{-}3, t, t\text{+}3\}$. Once trained, the 512-dimensional deep speaker features are derived from the output of the penultimate hidden layer, and the utterance-level d-vectors are obtained by average pooling. Similarly, the d-vectors are transformed to 150-dimensional vectors by LDA, and the PLDA model is employed to score the trials. The hyper-parameter of the Gaussian-constrained training is empirically set to 0.01.

4.3. Results

4.3.1. SITW

The results on the two SITW evaluation sets, *Dev. Core* and *Eval. Core*, are reported in Table 1 and Table 2, respectively. The results are reported in terms of three metrics: the equal error rate (EER), and the minimum of the normalized detection cost function (minDCF) with two settings: one with the prior target probability P_{tar} set to 0.01 (DCF(10^{-2})), and the other with P_{tar} set to 0.001 (DCF(10^{-3})).

From these results, it can be observed that the proposed Gaussian-constrained training improves both the x-vector and d-vector systems, in terms of all these three metrics. Furthermore, this approach seems more effective for the x-vector system. A possible reason is that for the d-vector system, the average pooling may corrupt the Gaussian property. Another possible reason is that the frame-level constraint in the d-vector system may lead to unstable parameter update compared to the utterance-level constraint in the x-vector system. Nevertheless, more investigation is required to under-

stand this discrepancy.

Table 1. Performance on SITW Dev. Core.

Embedding	DCF(10^{-2})	DCF(10^{-3})	EER(%)
i-vector	0.4279	0.5734	4.967
d-vector	0.4875	0.6837	5.314
d-vector + *Gauss*	0.4861	0.6812	5.160
x-vector	0.3025	0.4862	2.965
x-vector + *Gauss*	**0.2826**	**0.4551**	**2.734**

Table 2. Performance on SITW Eval. Core.

Embedding	DCF(10^{-2})	DCF(10^{-3})	EER(%)
i-vector	0.4577	0.6214	5.249
d-vector	0.5206	0.7570	5.686
d-vector + *Gauss*	0.5149	0.7496	5.659
x-vector	0.3235	0.4875	3.390
x-vector + *Gauss*	**0.3032**	**0.4520**	**3.034**

4.3.2. CSLT-SITW

The performance on the CSLT-SITW set is reported in Table 3. Note that the acoustic properties and linguistic conditions are clearly different from the training data. From Table 3, it can be observed that in spite of this mismatch, the Gaussian-constrained training still delivers consistent performance improvement on both the x-vector and d-vector systems, at least in terms of EER and DCF(10^{-2}). The strange degradation in DCF(10^{-3}) may be attributed to the fact that this new training approach emphasizes on a different operation point, though more analysis is required.

Table 3. Performance on CSLT-SITW.

Embedding	DCF(10^{-2})	DCF(10^{-3})	EER(%)
i-vector	0.4425	0.5698	6.479
d-vector	0.3881	0.4584	5.494
d-vector + *Gauss*	0.3706	0.4701	5.297
x-vector	0.2731	**0.3227**	4.139
x-vector + *Gauss*	**0.2418**	0.3746	**3.474**

5. CONCLUSIONS

This paper proposed a Gaussian-constrained training that can be applied to both the feature-based system (d-vector) and utterance-based system (x-vector) for ASV. The basic idea is to enforce a parameter-free classifier so that all the knowledge of the training data would be learned by the feature component; additionally, it encourages the derived speaker features, at either frame-level or utterance-level, to be Gaussian. The former allows more effective usage of the training data, and the latter boosts the PLDA scoring. The experimental results demonstrated that the proposed approach can deliver consistent performance improvement, not only on matched data, but also on non-matching conditions. As for the future work, more comprehensive analysis will be conducted to understand the behavior of the Gaussian-constrained training, e.g., the impact of the constraint on different layers.

6. REFERENCES

[1] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre 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.

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

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*. IEEE, 2014, pp. 4052–4056.

[5] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *ICASSP*. IEEE, 2016, pp. 5115–5119.

[6] Lantian Li, Yixiang Chen, Ying Shi, Zhiyuan Tang, and Dong Wang, "Deep speaker feature learning for text-independent speaker verification," in *Interspeech*, 2017, pp. 1542–1546.

[7] Sergey Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision–ECCV*, pp. 531–542, 2006.

[8] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Odyssey*, 2014, pp. 293–298.

[9] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*. IEEE, 2014, pp. 1695–1699.

[10] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[11] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*. IEEE, 2018.

[13] Lantian Li, Zhiyuan Tang, Dong Wang, and Thomas Fang Zheng, "Full-info training for deep speaker feature learning," in *ICASSP*. IEEE, 2018, pp. 5369–5373.

[14] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[15] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[16] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*. IEEE, 2017, pp. 5220–5224.

[17] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The Speakers in the Wild (SITW) speaker recognition database.," in *Interspeech*, 2016, pp. 818–822.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[19] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Interspeech*, 2017, pp. 999–1003.