Team 55: SDSV challenge Task 2 - Text Independent Speaker Verification: A Technical Report

Shreyas Ramoji, Prashant Krishnan, Sriram Ganapathy

Learning and Extraction of Acoustic Patterns (LEAP) Lab, Department of Electrical Engineering
Indian Institute of Science, Bengaluru, India
{shreyasr, prashantkv1, sriramg}@iisc.ac.in

Abstract

In this report, we provide a description of our systems submitted to submitted to the Short Duration Speaker Verification Challenge Task 2 - Text independent speaker Verification. Our systems are based on an x-vector model trained on the VoxCeleb dataset with Gaussian PLDA and Discriminative Neural PLDA backends trained on VoxCeleb and SDSVC Track 2 train datasets. With the NPLDA, we observed significant 28% relative improvement over the regular GPLDA model.

1. Introduction

The development of i-vectors as fixed dimensional front-end features for speaker recognition and verification tasks was introduced in [1, 2]. In the recent years, neural network embeddings trained on a speaker discrimination task were proposed as features to replace the i-vectors. These features called x-vectors [3] were shown to improve over the i-vectors for speaker recognition [4].

Following the extraction of x-vectors/i-vectors, different pre-processing steps are employed to transform the embeddings. The common steps include linear discriminant analysis (LDA) [2], unit length normalization [5] and within-class covariance normalization (WCCN) [6]. The transformed vectors are modeled with probabilistic linear discriminant analysis (PLDA) [7]. The PLDA model is used to compute a log likelihood ratio from a pair of enrollment and test embeddings which is used to verify whether the given trial is a target or non-target.

Recently in [8, 9], we proposed a neural backend model which jointly performs pre-processing and scoring. This model, refered to as neural PLDA (NPLDA), operates on pairs of x-vector embeddings (a pair of enrollment and test x-vectors), and outputs a score that allows the decision of target versus non-target hypotheses. The implementation using neural layers allows the entire model to be learnt using a speaker verification cost. The use of conventional cost functions like binary cross entropy cause overfitting of the model to the training speakers, and have generalization issues on evaluation sets. In an attempt to avoid this, we use an approximation to the minimum detection cost (minDCF) [10] to optimize the neural backend model.

The rest of the paper is organized as follows. In Section 2, we describe the front-end configurations used for feature processing and x-vector extraction. Section 3 describes the proposed neural network architecture used, and the connection with generative PLDA model and the cost function which is used as an objective to optimize the neural network. This is followed by discussion of experiments and results in Section 4 and a brief set of concluding remarks in Section 5.

2. Front-end Model - X-vector Extractor

In this section, we provide the description of the front-end feature extraction and x-vector model configuration.

2.1. Training

The x-vector extractor is trained entirely using speech data extracted from combined VoxCeleb 1 [11] and VoxCeleb 2 corpora [12]. These datasets contain speech extracted from celebrity interview videos available on YouTube, spanning a wide range of different ethnicities, accents, professions, and ages. For training the x-vector extractor, we use $1,276,888$ segments from 7323 speakers selected from Vox-Celeb 1 (dev and test), and VoxCeleb 2 (dev).

This x-vector extractor was trained using 30 dimensional Mel-Frequency Cepstral Coefficients (MFCCs) from 25 ms frames shifted every 10 ms using a 23-channel mel-scale filterbank spanning the frequency range 20 Hz - 7600 Hz. A 5-fold augmentation strategy is used that adds four corrupted copies of the original recordings to the training list [3, 4]. The augmentation step generates $6,384,440$ training segments for the combined VoxCeleb set.

An extended TDNN with 12 hidden layers and rectified linear unit (RELU) non-linearities is trained to discriminate among the nearly 7000 speakers in the training set [4]. The first 10 hidden layers operate at frame-level, while the last 2 layers operate at segment-level. There is a 1500-dimensional statistics pooling layer between the frame-level and segment-level layers that accumulates all frame-level outputs using mean and standard deviation. After training, embeddings are extracted from the 512 dimensional affine component of the 11th layer (i.e., the first segment-level layer). More details regarding the DNN architecture and the training process can be found in [4].

3. The Neural PLDA Backend

In the proposed pairwise discriminative network (NPLDA) (Fig. 1), we construct the pre-processing steps of LDA as first affine layer, unit-length normalization as a non-linear activation and PLDA centering and diagonalization as another affine transformation. The PLDA pair-wise scoring function is implemented as a Quadratic layer in Fig. 1. Thus, the NPLDA implements the pre-processing of the x-vectors and the PLDA scoring as a neural backend. The parameters of the NPLDA model are initialized with the baseline system and these parameters are learnt in a backpropagation setting.

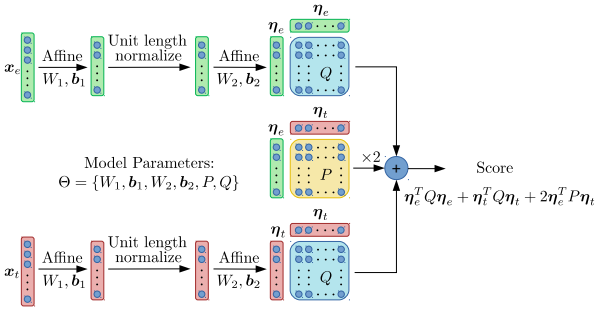

Figure 1: *Neural PLDA Net Architecture: The two inputs $\boldsymbol{x}_e$ and $\boldsymbol{x}_t$ are the enrollment and test x-vectors respectively.*

3.1. Soft Detection Cost

The normalized detection cost function (DCF) [10] is defined as:

$$C_{Norm}(\beta, \theta) = P_{Miss}(\theta) + \beta P_{FA}(\theta) \tag{1}$$

where β is an application based weight defined as

$$\beta = \frac{C_{FA}(1 - P_{target})}{C_{Miss} P_{target}} \tag{2}$$

where C_{Miss} and C_{FA} are the costs assigned to miss and false alarms, and P_{target} is the prior probability of a target trial. P_{Miss} and P_{FA} are the probability of miss and false alarms respectively, and are computed by applying a detection threshold of θ to the log-likelihood ratios.

$$P_{Miss}(\theta) = \frac{\sum_{i=1}^{N} t_i \mathbb{1}(s_i < \theta)}{\sum_{i=1}^{N} t_i} \tag{3}$$

$$P_{FA}(\theta) = \frac{\sum_{i=1}^{N} (1 - t_i) \mathbb{1}(s_i \geq \theta)}{\sum_{i=1}^{N} (1 - t_i)}. \tag{4}$$

Here, s_i is the score (LLR) output by the model, t_i is the ground truth variable for trial i. That is, $t_i = 0$ if trial i is a target trial, and $t_i = 1$ if it is a non-target trial. $\mathbb{1}$ is the indicator function. The normalized detection cost function (eq. 1) is not a smooth function of the parameters due to the step discontinuity induced by the indicator function $\mathbb{1}$, and hence, it cannot be used as an objective function in a neural network. We propose a differentiable approximation of the normalized detection cost by approximating the indicator function with a sigmoid function.

$$P_{Miss}^{(\text{soft})}(\theta) = \frac{\sum_{i=1}^{N} t_i \left[1 - \boldsymbol{\Sigma}(\alpha(s_i - \theta))\right]}{\sum_{i=1}^{N} t_i} \tag{5}$$

$$P_{FA}^{(\text{soft})}(\theta) = \frac{\sum_{i=1}^{N} (1 - t_i) \boldsymbol{\Sigma}(\alpha(s_i - \theta))}{\sum_{i=1}^{N} (1 - t_i)} \tag{6}$$

By choosing a large enough value for the warping factor α, the approximation can be made arbitrarily close to the actual detection cost function for a wide range of thresholds.

The primary cost metric of the SDSVC challenge is the normalized detection cost function (actDCF) computed at the threshold of $\log \beta$ applied to the LLRs. This is given by

$$C_{Primary} = C_{Norm}(\beta), \log \beta_e) \tag{7}$$

where $\beta = 9.9$. We compute the Neural PLDA loss function as

$$\mathcal{L}_{Primary} = C_{Norm}^{(\text{soft})}(\beta, \theta) \tag{8}$$

where θ is the thresholds which minimizes $\mathcal{L}_{Primary}$ when included as a model parameter. The minimum detection cost (minDCF) is achieved at a threshold where the DCF is minimized.

$$\text{minDCF} = \min_{\theta} C_{Norm}(\beta, \theta) \tag{9}$$

In other words, it is the best cost that can be achieved through calibration of the scores. We include these thresholds in the set of parameters that the neural network learns to minimize minDCF through backpropagation of the soft detection cost function $\mathcal{L}_{Primary}$.

4. Experiments and Results

We perform several experiments with the proposed neural net architecture and compare them with various discriminative backends previously proposed in the literature such as the discriminative PLDA [13, 14] and pairwise Gaussian backend [15]. We also compare the performance with the baseline system using Kaldi recipe that implements the generative PLDA model based scoring.

For all the pairwise generative/discriminative models, we train the backend using randomly sampled target and non-target pairs which are matched by gender. We perform experiments by sampling trials from the clean VoxCeleb segments, and also the augmented set. We sample about 6.6 million trials from the clean set and around 33 million trials from the augmented set.

4.1. GPLDA Baseline

The primary baseline to benchmark our systems is the generative Gaussian PLDA (GPLDA) backend implementation in the Kaldi toolkit. The Kaldi implementation models the average embedding x-vector of each training speaker. The x-vectors are centered, dimensionality reduced using LDA to 170 dimensions, followed by unit length normalization. The linear transformations and the GPLDA matrices are used to initialize the proposed pairwise PLDA network.

4.2. Neural PLDA (NPLDA)

Before the gender labels of the train dataset were released, we trained a gender classifier using the VoxCeleb dataset to assign gender labels to the SDSVC Task-2 train set. We then manually generated gender matched target and non-target trials from the VoxCeleb and SDSVC datasets separately. All the trials were pooled into batches, split into batches of size 2048 trials, which were used to train the NPLDA model. We initialized the model with the pretrained GPLDA from Kaldi, and optimized the NPLDA network with the soft detection cost function. This is our primary submission which is also the single best system.

Model	PLDA Train Dataset	MinDCF on SDSVC progress set
GPLDA	VoxCeleb	0.386
GPLDA	VoxCeleb Aug	0.354
GPLDA	SDSVC Task 2 - Train	0.352
NPLDA	VoxCeleb + SDSVC Task 2 - Train	0.253

Table 1: *Performances of Submitted Systems*

5. Summary and Conclusions

By participating in THE SDSVC challenge, we experimented with the use of NPLDA as a backend for Speaker Verification in the SDSV Challenge with a regular old x-vector model. This gave us a significant 28% relative improvement over the regular GPLDA model.

6. References

[1] 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.

[2] 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.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP 2018*. IEEE, 2018, pp. 5329–5333.

[4] M. Mclaren, D. Castán, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 327–334.

[5] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech 2011*, 2011.

[6] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Ninth international conference on spoken language processing*, 2006.

[7] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, pp. 14–21.

[8] S. Ramoji, P. Krishnan, and S. Ganapathy, "NPLDA: A Deep Neural PLDA Model for Speaker Verification," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 202–209. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-29

[9] S. Ramoji, P. Krishnan, B. Mysore, P. Singh, and S. Ganapathy, "LEAP System for SRE 2019 CTS Challenge - Improvements and Error Analysis," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 281–288. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-40

[10] D. A. Van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker classification I*. Springer, 2007, pp. 330–353.

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

[12] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018*, 2018, pp. 1086–1090. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1929

[13] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *ICASSP 2011*. IEEE, 2011, pp. 4832–4835.

[14] O. Glembek, L. Burget, N. Brümmer, O. Plchot, and P. Matějka, "Discriminatively trained i-vector extractor for speaker verification," in *Interspeech 2011*, 2011.

[15] S. Cumani, N. Brümmer, L. Burget, P. Laface, O. Plchot, and V. Vasilakakis, "Pairwise discriminative speaker verification in the i-vector space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1217–1227, 2013.