Statistics Pooling Time Delay Neural Network Based on X-vector for Speaker Verification

Qian-Bei Hong[1], Chung-Hsien Wu[1,2], Hsin-Min Wang[1], and Chien-Lin Huang[3]

[1]Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Tainan, Taiwan
[2]Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan
[3]PingAn AI Lab, Palo Alto, CA 94306, USA

Outline

- Introduction
- Related Work
- Frame-Level Statistics Pooling TDNN
- Experimental Results
- Conclusions

Introduction

- Recently, deep neural networks (DNN) have been widely applied to capture speaker characteristics and produce speaker embedding as speaker representation in speaker verification (SV) tasks.
  - Bottleneck feature, d-vector, x-vector, and so on.

- Most SV systems are based on x-vector features.
  - The architecture consists of two feature transformations.
    - Frame-level feature transformation
    - Segment-level feature transformation

Introduction

- Many studies are focused on improving performance for speaker verification by adding various layers or considering the contributions from different models.
  - Attention mechanism
  - Model-level fusion

- This paper aims to improve speaker embedding representation based on x-vector for extracting more detailed information.

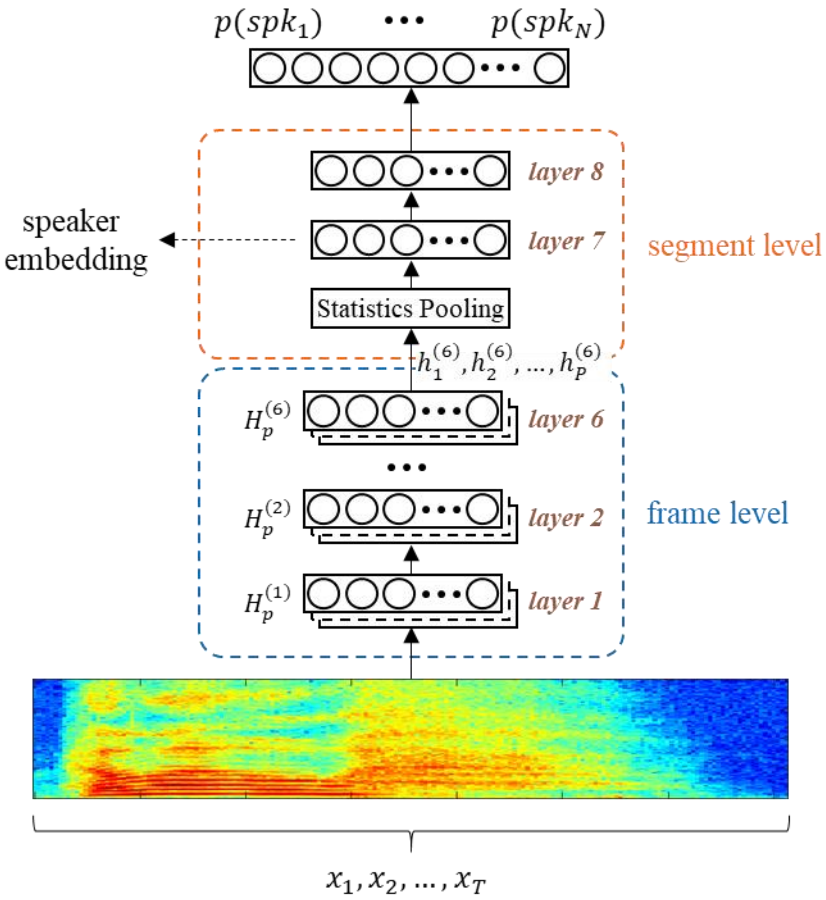
Related Work

- The x-vector embedding with PLDA classifier is the SOTA system for speaker verification.

- In the TDNN, given a subsequence of F output vectors $H_p^{l-1} = \{h_{p,1}^{l-1}, h_{p,2}^{l-1}, \dots, h_{p,F}^{l-1}\}$ from the previous $(l-1)$th layer at time step p

$$h_p^l = \alpha\left(W^l H_p^{l-1} + b^l\right) \qquad (1)$$

where $W^l \in \mathbb{R}^{D^l \times Q^l}$ is the weight matrix of size $D^l \times Q^l$, D^l is the number of output nodes and Q^l is the number of input nodes; b^l is the bias vector in layer l and $\alpha(\cdot)$ is the activation function.



Problems

- As the TDNN layer focuses on local feature extraction
  - High-level feature extraction through non-linear transformations with low weights in preceding layers may lose some important information using low-level features.

- In real-world environment applications, sometimes there are multi-speakers talking at the same time.
  - The embeddings extracted from multi-speaker recordings will cause the confusion of speaker characteristics and decrease the recognition performance.

Goals

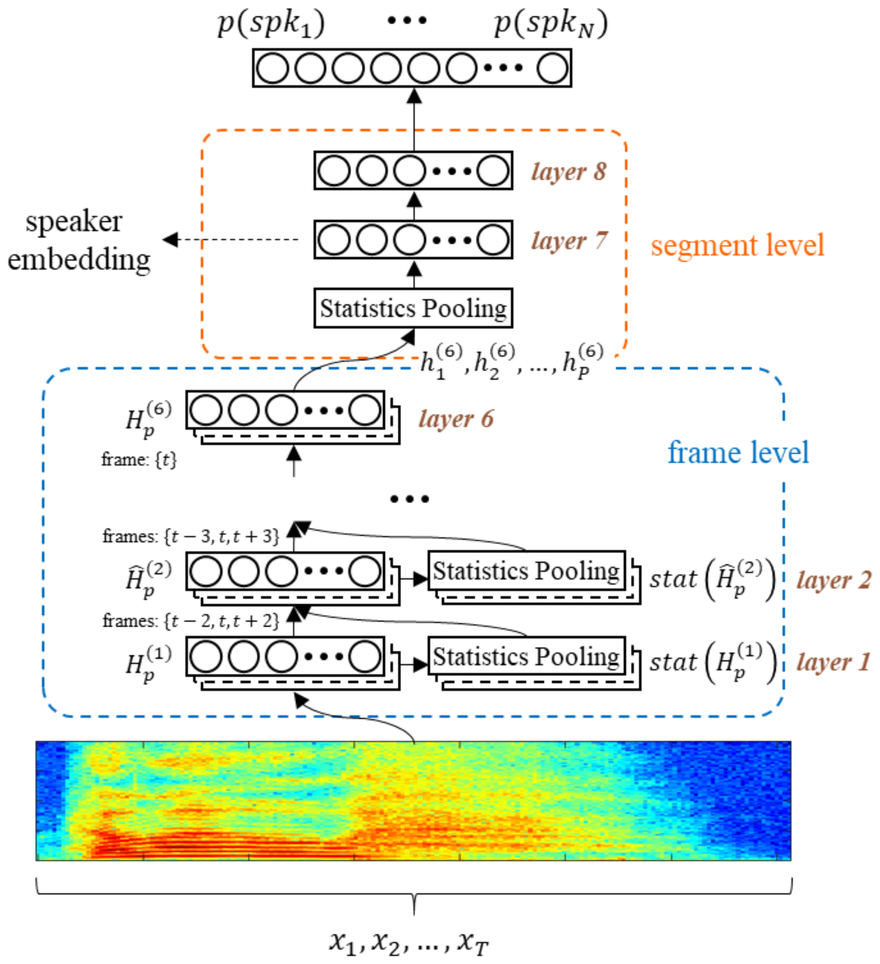
- This study integrates TDNN with the statistics pooling to exploit the potential of the network by considering the variation of temporal context.
  - To improve the ability of x-vector learning by capturing more robust speaker characteristics.
  - To reduce the interference from other speakers in the recordings.

Frame-Level Statistics Pooling TDNN

- We directly combine H_p^{l-1} and statistics pooling result of H_p^{l-1} to form a new input feature vector, which is then fed into the next layer

$$\hat{h}_p^l = \alpha\left(W^l\left[H_p^{l-1} \oplus stat\left(H_p^{l-1}\right)\right] + b^l\right) \quad (2)$$

where $\oplus$ denotes a concatenation operation, $stat(\cdot)$ is the statistics pooling function that computes the mean and standard deviation.



Frame-Level Statistics Pooling TDNN

- Assuming that the input is stationary speech, each output vector is similar to the other output vectors. The transformation can thus be simplified as follows.

$mean(E[\hat{H}^{l-1}]) = \hat{h}_{p,1}^{l-1} = \hat{h}_{p,2}^{l-1} = \cdots = \hat{h}_{p,F}^{l-1}$

$E[\hat{H}^{l-1}] = \hat{H}_1^{l-1} = \hat{H}_2^{l-1} = \cdots = \hat{H}_P^{l-1}$

$std(\cdot) = [0,0,\ldots,0]$

$$\hat{H}_p^{l-1} = \{\hat{h}_{p,1}^{l-1}, \hat{h}_{p,2}^{l-1}, \ldots, \hat{h}_{p,F}^{l-1}\} \quad (3)$$

$$\hat{H}^{l-1} = \{\hat{H}_1^{l-1}, \hat{H}_2^{l-1}, \ldots, \hat{H}_P^{l-1}\} \quad (4)$$

$$\begin{aligned} \hat{h}_p^l &\approx \alpha\left(W^l\left[E\left[\hat{H}^{l-1}\right] \oplus mean\left(E\left[\hat{H}^{l-1}\right]\right) \oplus std\left(E\left[\hat{H}^{l-1}\right]\right)\right] + b^l\right) \\ &\approx \underline{\alpha\left(W^l\left[\hat{H}_p^{l-1} \oplus \hat{h}_{p,f}^{l-1} \oplus std\left(\hat{H}_p^{l-1}\right)\right] + b^l\right)} \quad (5) \end{aligned}$$

$\approx \alpha\left(W^l \hat{H}_p^{l-1} + b^l\right) = h_p^l$

where $\hat{H}^{l-1}$ is a set of subsequences corresponding to P time steps obtained from the previous $(l-1)$th layer, $mean(\cdot)$ is the mean function and $std(\cdot)$ is the standard deviation function.

Datasets

- Training data:
  - VoxCeleb2 dataset
    - The *DEV* set contained 1,092,009 utterances from 5,994 celebrities, which were obtained from YouTube videos.
- Testing data:
  - VoxCeleb1 dataset
    - The dataset contained 153,516 utterances from 1,251 celebrities, which was also obtained from YouTube videos.
  - The Speakers in the Wild (SITW) dataset
    - The *EVAL* dataset provides 2,883 recordings from 180 speakers, which contained multi-speaker presentations in the same utterances.

Experimental Setup

- Input features
  - 40-dimentional Mel-frequency cepstral coefficients (MFCCs)
  - the spectrogram is extracted from a 25ms window with a stride of 10ms.
- In the following results
  - ***"x-vector"*** refers to baseline system using x-vector
  - ***"stats-vector"*** refers to the system using the proposed feature representation
  - ***"fusion"*** refers to the score fusion method

$$scoreF_i = \frac{1}{K}\sum_{k=1}^{K}\left(score_i(k) - \frac{1}{S}\sum_{s=1}^{S} score_s(k)\right) + \frac{1}{KS}\sum_{k=1}^{K}\sum_{s=1}^{S} score_s(k) \qquad (6)$$

where K is the number of speaker verification systems, S is the number of embedding pairs, and $scoreF_i$ is the i-th score that was determined by the average score of each system and total average score of all systems.

Experimental Results

- Evaluation on VoxCeleb1

Table 1. Results on the VoxCeleb1.

System	VoxCeleb1 (cleaned)			VoxCeleb1-E (cleaned)			VoxCeleb1-H (cleaned)		
	EER	$DCF10^{-2}$	$DCF10^{-3}$	EER	$DCF10^{-2}$	$DCF10^{-3}$	EER	$DCF10^{-2}$	$DCF10^{-3}$
x-vector	3.50	0.4009	0.6012	3.45	0.3915	0.6248	6.02	0.5387	0.7740
stats-vector	3.29	0.3633	**0.4820**	3.39	0.3844	0.6276	5.94	0.5439	0.7849
fusion	**2.96**	**0.3542**	0.5238	**3.11**	**0.3629**	**0.6065**	**5.48**	**0.5184**	**0.7597**

Compared to the baseline x-vector system, the stats-vector system performed better by 6.0%, 1.7% and 1.3% in EER, respectively.

Using score fusion significantly improved the performances
>> VoxCeleb1 (cleaned): improved by 15.4% in EER and 11.6% in $DCF10^{-2}$.

Experimental Results

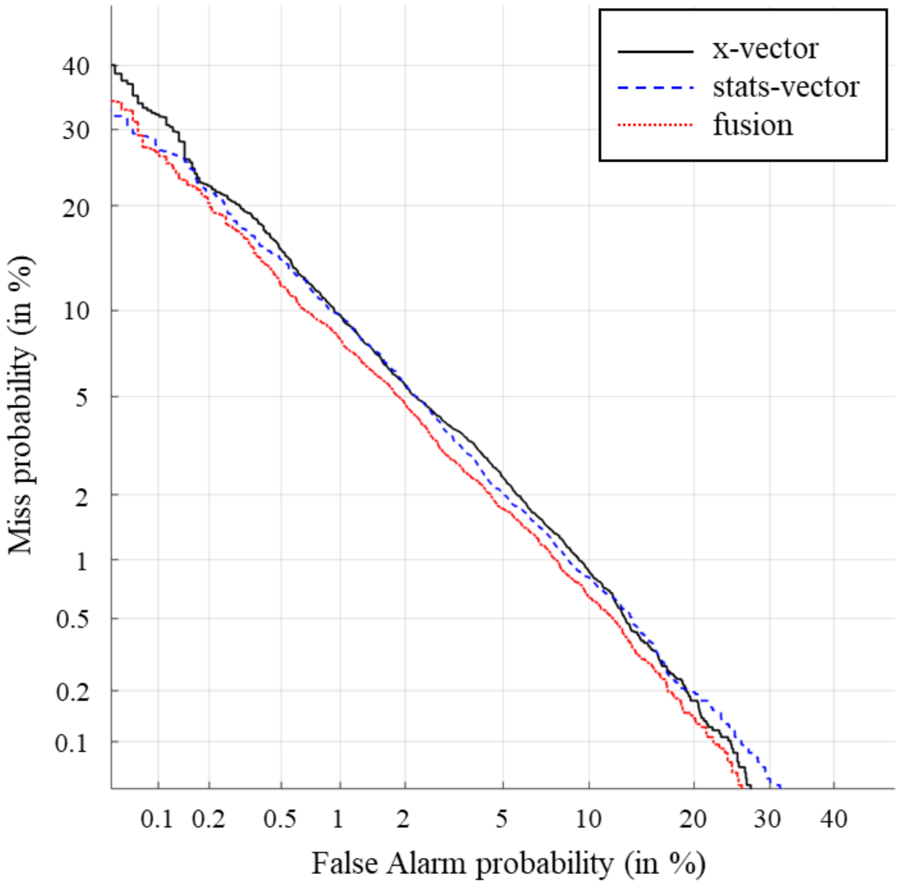
- Evaluation on SITW

Table 2. Results on the SITW EVAL set.

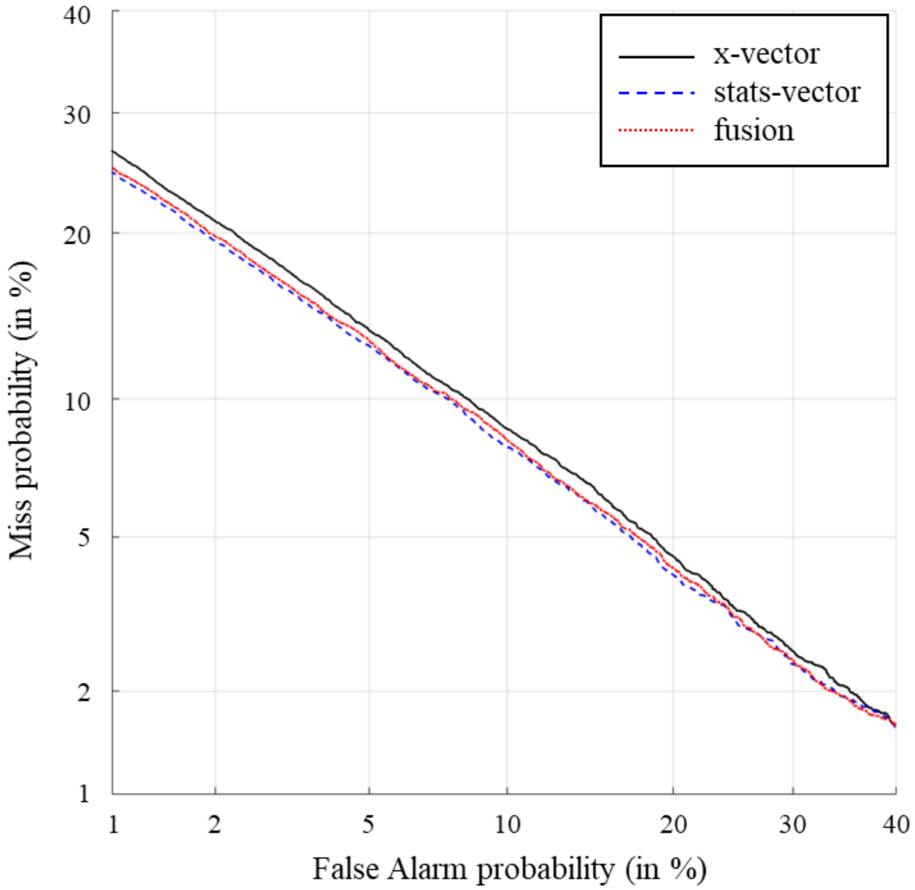
System	EVAL core-core			EVAL core-multi			EVAL assist-core			EVAL assist-multi		
	EER	$DCF10^{-2}$	$DCF10^{-3}$	EER	$DCF10^{-2}$	$DCF10^{-3}$	EER	$DCF10^{-2}$	$DCF10^{-3}$	EER	$DCF10^{-2}$	$DCF10^{-3}$
x-vector	4.87	0.4691	0.7023	7.72	0.5635	0.7744	7.67	0.5134	0.7279	9.22	0.5705	0.7859
stats-vector	4.74	0.4506	**0.6635**	**7.37**	**0.5427**	**0.7524**	**7.31**	**0.4987**	**0.6835**	**8.78**	**0.5507**	**0.7493**
fusion	**4.69**	**0.4495**	0.6773	7.44	0.5450	0.7581	7.43	0.5005	0.7014	8.97	0.5545	0.7627

The stats-vector obtained the best performance on *EVAL* assist-multi trial list, outperforming by 4.8% in EER and 3.5% in $DCF10^{-2}$.

Experimental Results



DET curve for the trial pairs in **VoxCeleb1 (cleaned)**



DET curve for the trial pairs in **SITW EVAL assist-multi**

Experimental Results

- In this study, compared to the baseline x-vector system
  - Evaluation on VoxCeleb1
    - EER (in VoxCeleb1 (cleaned)): 3.50% -> 3.29%
  - Evaluation on SITW
    - EER (in *EVAL* assist-multi): 9.22% -> 8.78%

- The proposed stats-vector system can significantly improve the speaker verification performance by considering the variation of temporal context in frame-level TDNN.

Conclusions

- This paper proposes a statistics pooling TDNN architecture (named as stats-vector) for speaker verification.
  - The TDNN structure integrates statistics pooling for each layer, to consider the variation of temporal context in frame-level transformation.
  - Compared to the x-vector architecture, this study only changed three layers in the frame-level transformation which could improve the performance of speaker verification.

Future Work

- Recently, statistics pooling replaced by attention mechanism has been proven that providing different speaker discriminative information of frames can achieve better performance.

  - In the future, we will investigate the potential of attention mechanisms, to further consider the different characteristics and improve the performance.

Thank you for your attention