SHORT-TIME SPECTRAL AGGREGATION FOR SPEAKER EMBEDDING

Youzhi Tu and Man-Wai Mak
The Hong Kong Polytechnic University
Department of Electronic and Information Engineering
Hong Kong SAR

ABSTRACT
State-of-the-art speaker verification systems take frame-level acoustics features as input and produce fixed-dimensional embeddings as utterance-level representations. Thus, how to aggregate information from frame-level features is vital for achieving high performance. This paper introduces short-time spectral pooling (STSP) for better aggregation of frame-level information. STSP transforms the temporal feature maps of a speaker embedding network into the spectral domain and extracts the lowest spectral components of the averaged spectrograms for aggregation. Benefiting from the low-pass characteristic of the averaged spectrograms, STSP is able to preserve most of the speaker information in the feature maps using a few spectral components only. We show that statistics pooling is a special case of STSP where only the DC spectral components are used. Experiments on VoxCeleb1 and VOiCES 2019 show that STSP outperforms statistics pooling and multi-head attentive pooling, which suggests that leveraging more spectral information in the CNN feature maps can produce highly discriminative speaker embeddings.

Index Terms— Speaker verification, speaker embedding, spectral pooling, statistics pooling

1. INTRODUCTION
Modern speaker verification (SV) systems typically comprise a front-end embedding extractor and a backend scorer. Traditional front-end systems use the i-vector method [1]. With the advance of deep neural networks (DNNs), DNN-based front-ends have gained increasing popularity. Among these techniques, the convolutional neural network (CNN) based embedding has been widely used in the SV community [2–5].

A classical DNN speaker embedding is the x-vector, which uses time delay neural networks (TDNNs) to extract frame-level features and applies statistics pooling to summarize these features in the form of fixed-length vectors. A TDNN can be viewed as a 1-D dilated CNN where only specific contextual frames within a sliding window are involved in the convolution operation. It has been shown that the x-vectors are more speaker discriminative than the i-vectors. They are also more robust to noise, reverberation, and domain mismatch [6–9]. There are also embedding systems that use ResNets [10] or DenseNets [11] for frame-level processing. For example, lightweight ResNets were adapted from ResNet-34 and ResNet-50 in [12] and [13]. In [5], a deep embedding network was implemented based on DenseNets.

The above speaker embedding extractors share a similar structure: a CNN-based frame-level network, a pooling layer, and a fully-connected utterance-level network. Because the frame-level network aims to produce fixed-dimensional embeddings from variable-length speech segments, how to aggregate speaker information from frame-level representations into utterance-level embeddings is of special importance. Because some speaker information will inevitably be lost in the aggregation process, it is essential to preserve as much information as possible during frame aggregation.

From a Fourier perspective, the conventional statistics pooling only exploits the DC (zero frequency) components in the frequency domain (see Section 3.3). An intuitive improvement is to use more frequency components besides the DC ones to retain richer information during aggregation. In [14], spectral pooling was proposed based on discrete Fourier transform (DFT) to retain rich information. Inspired by this idea, we proposed to perform aggregation by transforming the temporal feature maps in the last convolutional layer of a speaker embedding network into the spectral domain. Because spectral representations have a desirable low-pass characteristic, it facilitates the aggregation in the transformed domain. From a signal processing perspective, the low-pass characteristic allows us to reconstruct the temporal features using a few low-frequency components with minimum distortion, because most of the energy of the temporal features can be computed from these components. Also, it ensures an ordered selection of the informative spectral components for pooling by merely retaining the lowest several components without the need for an attention mechanism to determine which components are informative.

However, because DFT can only be applied to deterministic or wide-sense stationary signals, it may not be suitable for non-stationary speech signals [15]. To account for the inher-
ent non-stationarity of channel-wise convolutional features, we replaced DFT with short-time Fourier transform (STFT) [16] and proposed short-time spectral pooling (STSP) for better preservation of speaker information.

2. RELATED WORKS

There have been various pooling methods for speaker embeddings. For example, early work [17] exploited the channel-wise mean vectors of the frame-level features through temporal pooling. The x-vector extractor further appended standard deviations through a statistics pooling layer [2], which has shown better performance compared with temporal pooling. Learnable dictionary encoding (LDE) was proposed in [18] and the encoded vectors act like the means of a Gaussian mixture model (GMM).

Another popular category is the attention-based pooling. In [19], an attention mechanism was introduced to weight the temporal frames so that the attended frames have large influence on the network outputs. Following the same strategy, [20] proposed multi-head attentive pooling to extract more speaker related information from the convolutional layer’s output. In [5], a mixture of attentive pooling was proposed from a probabilistic point of view, integrating the attention mechanism and GMM clustering. This method has been shown to outperform multi-head attentive pooling on VoxCeleb1 and VOiCES 2019.

In [13], a joint time-frequency pooling was proposed to aggregate information in 3D tensors. However, the pooling operations are applied independently on the frequency axis and the time axis. Therefore, the pooling in [13] is different from our proposed method in Section 3.3, where pooling is operated in the transformed domain. Thus, we used the term spectral pooling to differentiate the proposed method from the frequency pooling in [13].

3. SHORT-TIME SPECTRAL AGGREGATION

In this section, we introduce some background knowledge and the principle of short-time spectral pooling.

3.1. Short-time Fourier Transform

Fourier analysis has been widely used in speech processing. Fourier transform converts a signal into the frequency domain by mapping it onto a series of sinusoidal and cosinusoidal bases. For a discrete sequence \( x = \{x(0), \ldots, x(N - 1)\} \in \mathbb{R}^N \), its DFT \( X \triangleq \mathcal{F}(x) = \{X(0), \ldots, X(N - 1)\} \in \mathbb{C}^N \) is defined as \( X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn} \), where \( k \in [0, N - 1] \).

However, DFT is not applicable to non-stationary signals like speech. To facilitate the spectral analysis of speech signals, STFT can be used to exploit their local stationarity. In fact, STFT can be seen as the Fourier transform of a windowed segment. Taking the \( c \)-th channel feature \( x_c \) of a convolutional feature map as an example, its STFT is:

\[
X_c(m, k) = \sum_{n=-\infty}^{\infty} x_c(n)w(n - mS)e^{-j\frac{2\pi}{L}kn},
\]

where \( w(\cdot) \) is a window function of length \( L \), \( S \) denotes the sliding step of the window, \( m \) indexes the sliding steps, and \( k = 0, \ldots, L - 1 \) indexes the frequency components.\(^1\) Eq. 1 suggests that by sliding the window, we may apply multiple STFTs on a 1D sequence to produce a 2D spectral feature map (spectrogram).

3.2. Spectral Pooling

Spectral pooling was proposed for better preservation of information compared with the stride-based pooling methods such as max pooling [14]. In computer vision, spectral pooling involves three steps: (1) transforming the convolutional features from the spatial domain to the frequency domain by using DFT, (2) cropping and retaining the lower frequency components, and (3) performing inverse DFT on the cropped features to transform them back to the spatial domain.

Note that Step 2 requires the input to have a low-pass characteristic in the frequency domain. Although natural images meet this requirement, speech signals may not. However, evidence in Section 3.3 suggests that the averaged short-time spectrograms do show a low-pass characteristic. As a result, this requirement does not pose a problem to the aggregation of speaker information. In fact, it is the DFT that may cause detrimental effects on aggregation due to the non-stationarity in the CNN feature maps. As shown in the top-left plot in Figure 1, the temporal feature \( x_c \) of a randomly selected channel shows a strong non-stationary characteristic. To overcome this problem, we use STFT to exploit the local stationarity of the CNN feature maps.

3.3. Short-time Spectral Pooling (STSP)

As shown in Figure 1, STFT is performed on the output of the last convolutional layer along the temporal axis. For each channel, we obtain a 2D spectrogram after STFT (the phase information is discarded). To perform aggregation for each channel, we average the 2D spectrograms along the temporal direction to obtain an averaged spectral array, i.e.,

\[
\hat{X}_c(k) = \frac{1}{M} \sum_{m=0}^{M-1} |X_c(m, k)|, \quad k = 0, \ldots, L - 1
\]

where \( L \) is the window length and \( M \) is the number of windowed temporal segments. For a window of length \( L \), we

\(^1\)In this paper, we always make sure that the STFT length (the length of Fourier transform during STFT) is equal to the window length \( L \).
Fig. 1. Pipeline of short-time spectral pooling. The left-most matrix in the middle row shows a temporal feature map extracted from the last convolutional layer. The bottom-left spectrograms were produced by STFT with length $L = 16$, and the red boxes on top of the spectrogram denotes spectral arrays to be averaged (along the time axis). The actual values of $\hat{X}_c$ and $P_c$ after applying Eq. 2 and Eq. 3 are shown in the middle and the right-most maps in the middle row, respectively. The top three plots correspond to the row vectors in $x_c(n)$, $\hat{X}_c(k)$ and $P_c(k)$ in the red boxes, respectively. All the spectral features in the black box are concatenated to form the final utterance-level statistics (see Eq. 4 and Eq. 5 for details).

have $M = \text{floor}((N - L)/S) + 1$. Similarly, if we average the square of the spectrograms, we obtain the second-order spectral statistics:

$$P_c(k) = \frac{1}{M} \sum_{m=0}^{M-1} |X_c(m, k)|^2. \quad (3)$$

During aggregation, we concatenate $\hat{X}_c(0)$ and the square roots of the lowest $R$ components of $P_c(k)$ as the utterance-level representations of channel $c$:

$$z_c = \text{CONCAT} \left( \hat{X}_c(0), \sqrt{P_c(0)}, \ldots, \sqrt{P_c(R-1)} \right). \quad (4)$$

The final utterance-level features are produced by concatenating the spectral statistics of all channels, that is,

$$z = \text{CONCAT} (z_0, \ldots, z_{C-1}), \quad (5)$$

where $C$ is the number of channels.

One benefit of using the averaged representations $\hat{X}_c(k)$ and $P_c(k)$ is that they have a low-pass characteristic. This characteristic facilitates the aggregation by keeping the lowest spectral components only, because these components contain most of the energy of the original features.\(^2\)

In fact, STSP has a close relationship with statistics pooling [2]. For example, if we set $k = 0$ and use a rectangular window in Eq. 2 without any overlap between successive segments ($S = L$), the DC component $\hat{X}_c(0)$ approximates the mean of $x_c$. On the other hand, setting $k = 0$ in Eq. 3 resembles the power. In the extreme case where $S = L = 1$, we have $P_c(0) = \frac{1}{N} \sum_{n=0}^{N-1} x_c(n)^2$. This means that under this condition, using means and standard deviations for statistics pooling is an analogy to using the DC components ($\hat{X}_c(0)$ and $P_c(0)$ in Eq. 2 and Eq. 3) for STSP. Thus, we may consider STSP as a generalized statistics pooling. Because we have higher-frequency components ($P_c(k)$’s for $k > 0$) for pooling, we can preserve more information than statistics pooling.

### 4. EXPERIMENTAL SETUP

Statistics pooling [2], multi-head attentive pooling [20] and the proposed STSP were compared. We evaluated the performance on the VoxCeleb1 test set (clean) [4] and the VOiCES 2019 development and evaluation sets [22].

#### 4.1. Training of Speaker Embedding Extractor

For the evaluation of VOiCES 2019, both VoxCeleb1 development and VoxCeleb2 development data were used for
training, which amounts to 2,105,949 utterances from 7,185 speakers. Whereas only the VoxCeleb2 development subset (2,092,009 utterances from 5,984 speakers) was used as the training set for VoxCeleb1 evaluation. We followed the Kaldi’s VoxCeleb recipe to prepare the training data, i.e., using 40-dimensional filter bank features, performing energy-based voice activity detection, implementing augmentation (by adding reverberation, noise, music and babble to the original speech), and filtering out utterances with a duration less than 4 seconds.\footnote{https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb/v2.} Totally we had approximately twice the number of clean utterances for training the embedding network.

We used the standard x-vector extractor\cite{2} as the baseline. For systems that use multi-head attentive pooling, we used an attention network with 500 ReLU hidden nodes and $H$ linear output nodes, where $H$ is the number of attention heads. For STSP, we used a rectangular window with a length of 16. There is no overlap between successive windowed segments. The length of Fourier transform for each windowed segment was set to 16.

The additive margin softmax loss\cite{23} was used for training and the additive margin was set to 0.25. The mini-batch size was set to 128 and there are around 2,337 mini-batches in one epoch. Each mini-batch was created by randomly selecting speech segments of 2–4s from the training data. We used the stochastic gradient descent (SGD) optimizer with a momentum of 0.9. The initial learning rate was 0.1 and it was decayed by half at Epochs 40, 60, 80 and 90, respectively. Totally, the networks were trained for 100 epochs. Once training was completed, the speaker embedding was extracted from the affine output of the first embedding layer.

### 4.2. PLDA training

We used the Gaussian PLDA backend for both evaluation tasks. For VoxCeleb1, the PLDA model was trained on the x-vectors extracted from the clean utterances. For VOiCES 2019, we trained the backend using the concatenated speech with the same video session and used utterances augmented with reverberation and noise. Before PLDA training, the x-vectors were projected onto a 200-dimensional space by LDA for VoxCeleb1 and 150-dimensional space for VOiCES 2019, followed by whitening and length normalization. The LDA projection matrix was trained on the same dataset as for training the PLDA models. For VOiCES 2019, we also applied adaptive score normalization\cite{24}. The cohort was generated from the PLDA training data by selecting the longest two utterances of each speaker.

### 5. RESULTS AND DISCUSSIONS

Table 1 shows the performance of different systems on VoxCeleb1 (clean), VOiCES19-dev, and VOiCES19-eval. For VoxCeleb1, We can observe that both the attentive pooling and STSP systems outperform the baseline, whereas the STSP systems generally perform better than the multi-head attentive counterparts. The best result obtained by attentive pooling was obtained when the number of head was set to 2. Further increasing the number of heads shows a slight performance degradation. Also, we see that STSP achieves similar performance when the number of second-order spectral components is 2 or 3. These two setups significantly outperform the other configurations. However, including higher-frequency components has a detrimental effect on STSP, as can be seen in the case of $R = 4$. This may be because there are more variations or noises in the higher-frequency components.

Performance on the VOiCES 2019 is similar to that of VoxCeleb1, although the performance gap between STSP and attentive pooling becomes smaller. This suggests that STSP is able to aggregate more speaker information into the embeddings and that including more spectral components is beneficial for robust SV.

<table>
<thead>
<tr>
<th>Pooling</th>
<th>VoxCeleb1 EER</th>
<th>DCF</th>
<th>V19-dev EER</th>
<th>DCF</th>
<th>V19-eval EER</th>
<th>DCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>stats pooling</td>
<td>2.13</td>
<td>0.227</td>
<td>2.32</td>
<td>0.273</td>
<td>6.19</td>
<td>0.467</td>
</tr>
<tr>
<td>AP ($H=1$)</td>
<td>2.05</td>
<td>0.221</td>
<td>2.40</td>
<td>0.291</td>
<td>6.02</td>
<td>0.465</td>
</tr>
<tr>
<td>AP ($H=2$)</td>
<td>1.96</td>
<td>0.207</td>
<td>2.10</td>
<td>0.270</td>
<td>5.72</td>
<td>0.468</td>
</tr>
<tr>
<td>AP ($H=3$)</td>
<td>1.99</td>
<td>0.218</td>
<td>2.09</td>
<td>0.270</td>
<td>5.79</td>
<td>0.484</td>
</tr>
<tr>
<td>AP ($H=4$)</td>
<td>2.01</td>
<td>0.232</td>
<td>2.12</td>
<td>0.292</td>
<td>5.92</td>
<td>0.514</td>
</tr>
<tr>
<td>STSP ($R=1$)</td>
<td>2.17</td>
<td>0.221</td>
<td>2.25</td>
<td>0.280</td>
<td>6.20</td>
<td>0.469</td>
</tr>
<tr>
<td>STSP ($R=2$)</td>
<td>1.91</td>
<td>0.199</td>
<td>2.05</td>
<td>0.283</td>
<td>5.67</td>
<td>0.478</td>
</tr>
<tr>
<td>STSP ($R=3$)</td>
<td>1.82</td>
<td>0.210</td>
<td>2.16</td>
<td>0.266</td>
<td>5.76</td>
<td>0.473</td>
</tr>
<tr>
<td>STSP ($R=4$)</td>
<td>1.93</td>
<td>0.220</td>
<td>2.08</td>
<td>0.275</td>
<td>5.84</td>
<td>0.488</td>
</tr>
</tbody>
</table>

### 6. CONCLUSIONS

In this paper, we proposed a new pooling method for speaker embedding. Due to the low-pass characteristic of the averaged short-time spectrograms, STSP is able to aggregate both DC components and higher-frequency spectral information into the utterance-level representations, which helps to feed more discriminative information into the speaker embeddings. Results of evaluations on VoxCeleb1 and VOiCES 2019 show that STSP generally outperforms the multi-head attentive pooling, which suggests that it is beneficial to perform aggregation in the spectral domain for SV.
7. REFERENCES


