Event-related data conditioning for acoustic event classification

Yuanbo Hou, Dick Botteldooren
WAVES Research Group, Ghent University, Belgium
Yuanbo.Hou@UGent.be, Dick.Botteldooren@UGent.be

Abstract

Models based on diverse attention mechanisms have recently shined in tasks related to acoustic event classification (AEC). Among them, self-attention is often used in audio-only tasks to help the model recognize different acoustic events. Self-attention relies on the similarity between time frames, and uses global information from the whole segment to highlight specific features within a frame. In real life, information related to acoustic events will attenuate over time, which means the information within some frames around the event deserves more attention than distant time global information that may be unrelated to the event. This paper shows that self-attention may over-enhance certain segments of audio representations, and smooth out the boundaries between events representations and background noises. Hence, this paper proposes an event-related data conditioning (EDC) for AEC. EDC directly works on spectrograms. The idea of EDC is to adaptively select the frame-related attention range based on acoustic features, and gather the event-related local information to represent the frame. Experiments show that: 1) compared with spectrogram-based data augmentation methods and trainable feature weighting and self-attention, EDC outperforms them in both the original-size mode and the augmented mode; 2) EDC effectively gathers event-related local information and enhances boundaries between events and backgrounds, improving the performance of AEC.

Index Terms: Acoustic event classification, event-related data conditioning, SpecAugment, mixup, feature weighting

1. Introduction

Acoustic event classification (AEC) performs multi-label classification on audio clips, which aims to identify whether target acoustic events occur in the audio clip. AEC can help the agent understand its surrounding environment and what is happening. Classical applications of AEC include security surveillance [1], robot hearing [2], monitoring [3] and smart cities [4, 5].

Previous methods for AEC are often based on multi-layer perceptron architecture [6], convolutional neural networks (CNN) [7, 8], recurrent neural networks (RNN) [9], and convolutional recurrent neural networks (CRNN) [10]. Recently, motivated by the excellent performance of the attention-based Transformer [11] on sequence-related tasks, various attention mechanisms [12, 13, 14], especially self-attention, have been widely used in audio-related tasks [15]. Self-attention [11] shows a good balance between the ability to model long-term dependencies and computational efficiency [16]. Memory controlled self-attention is used to model relations between tokens within sound events to improve sound recognition [17]. And models with self-attention can be combined with connectionist temporal classification (CTC) loss to enforce a monotonic ordering and utilize timing information of acoustic events in audio clips [18]. CNN-Transformer is also proposed for audio event tasks, which performs similarly to the CRNN, but has the advantage of being efficiently trained in parallel [19].

In audio-related Transformer-based models [15, 20, 21, 22, 23], self-attention measures the cross-frame similarity within the whole segment and obtains new representations by weighted summation. In the above-mentioned models related to acoustic events, during the training, self-attention blindly transforms the frame based on global information from the whole segment, regardless of whether the information is related or unrelated to events in the frame. The globally calculated self-attention, also known as scaled dot-product attention [11], has the following drawbacks. 1) The information attenuation of sound events over time in real life is not considered. In real life, audio events usually last a few seconds. For some daily events, the shortest duration is about 0.5 s [24]. And events containing fluctuating or intermittent sounds (e.g., dogs barking, birds singing) usually consist of short events. That is, sound events in real life usually last a few seconds, and they are more related to other events in the neighboring time than events in a distant time. So the local information around the event deserves more attention. 2) After weighting and representing the frame based on global information, other events’ information that may be irrelevant to acoustic events in the frame will be added. That is, self-attention may not be able to effectively focus on the event-related local information, and cannot prevent the introduction of irrelevant information into representations of acoustic events. 3) Reconstructing local features based on global attention will make the local features approach global representations, resulting in boundaries between acoustic events and backgrounds being smoothed out in audio clips. It is difficult for the model to judge where the information of events is located, which increases the learning burden of the model. These shortcomings of self-attention are inconsistent with the characteristics of audio events. To mitigate the three deficiencies mentioned above, this paper proposes an event-related data conditioning (EDC) for AEC. EDC automatically selects the event-related attention range for each frame, locally focusing on related information without introducing features of unrelated distant time sounds into the representation.

The proposed data conditioning method EDC directly acts on spectrograms and does not require training, which is similar to spectrogram-based augmentation methods: SpecAugment [25] and Mixup [26]. SpecAugment introduces artificial noises on spectrograms to diversify training samples. Mixup randomly mixes spectrograms of two different samples in proportion to form a new sample. These data augmentation methods make the model focus on core features and ignore irrelevant details by creating diverse training samples with the same relevant features. The EDC will be compared with SpecAugment and Mixup. The essence of EDC is deterministic event-related feature transformation. And feature weighting [27], which is dedicated to dynamically adapting learnable features, also performs well in the AEC task. Therefore, this paper will also compare and demonstrate the performance of feature weighting. Even though EDC, Mixup, SpecAugment, and feature weighting are different approaches, their underlying goals are the same: to
make the model less sensitive to irrelevant features of target events and attempt to preserve key features. The proposed EDC is inspired by the performance analysis of self-attention in AEC tasks, so EDC that does not require training will be compared with the trainable self-attention to show the difference.

This paper proposes EDC, which is a deterministic feature transformation to gather event-related local information and enhance boundaries between events and backgrounds. This paper is organized as follows. Section 2 introduces the proposed EDC. Section 3 describes the dataset, model, experimental setup, and analyzes the results. Section 4 gives conclusions.

2. Proposed method
The acoustic event classification task, and the proposed event-related data conditioning method, are introduced in this part.

2.1. Acoustic event classification (AEC)
AEC performs multi-label classification on audio clips to identify and recognize target events [28]. In acoustic event-related tasks, the most frequently used efficient acoustic feature is the log mel spectrogram [6, 9, 19, 26, 29]. The input audio clip \( x \) denotes the acoustic time-frequency representation of the input audio clip, \( k \) the audio clip, and \( \alpha \) the weighting of offsets should decrease. This knowledge is simply introduced by

\[
\omega_i = \frac{1}{\sum_{j=1}^{T-i} \lambda_j} \quad \omega_j = \frac{1}{\sum_{j=T+1-i}^{T-i} \lambda_{T-j}}
\]

where the row vector \( \omega_i \in \mathbb{R}^{T \times F} \) denotes the acoustic time-frequency representation of the input audio clip, where \( \omega_i \in \mathbb{R}^{1 \times F} \) is the \( i \)-th frame, \( T \) and \( F \) denote the number of time frames and the acoustic feature dimension, respectively. For \( X \), the corresponding cross-frame similarity matrix [30] \( \Omega \) is obtained by:

\[
\Omega = XX^T = \begin{bmatrix} x_1 x_1 & \cdots & x_1 x_T \\ x_2 x_1 & \cdots & x_2 x_T \\ \vdots & \ddots & \vdots \\ x_T x_1 & \cdots & x_T x_T \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_T \end{bmatrix}
\]

where the row vector \( \omega_i \in \mathbb{R}^{T} \) is the similarity vector of the \( i \)-th frame. Specifically, \( \omega_i = [\lambda_1, \lambda_2, \ldots, \lambda_{T-i}, \lambda_{i+1}] \) is the similarity vector of the \( i \)-th frame. The cross-frame similarity matrix \( \Omega \) is truncated based on \( \Phi \) and normalized by row-wise softmax,

\[
EDC_{output} = softmax(\Omega \odot \Phi)X
\]

(8)

then \( P_{f_{\omega i}} \) is viewed as the probability of the frame-related similarity offset value distribution. The offset value of each previous frame position relative to the \( i \)-th frame and corresponding to \( f_{\omega i} \) is \( O_{f_{\omega i}} = \{i-1, i-2, \ldots, 1, 0\} \). Hence, based on the distribution probability of similarity at offset values, the expected value of the offset is

\[
\hat{f}_{\text{offset}} = E(O_{f_{\omega i}}) = \sum_{n=1}^{i} O_{f_{\omega i}}(n)P_{f_{\omega i}}(n)
\]

(6)

where the similarity value in \( [i - \hat{f}_{\text{offset}}, i] \) time axis is regarded as effective similar information related to the \( i \)-th frame. This definition of \( \hat{f}_{\text{offset}} \) is a mask for unwanted effects in the case of multiple similar events that cause the probability distribution to show multiple peaks. Therefore a further step is introduced.

3) Weight the effective similarity time range
As time increases, the probability that similar features belong to the same event decreases. Thus, independently of the probability of \( P_{f_{\omega i}} \), the weighting of offsets should also decrease. This knowledge is simply introduced by

\[
f_{\text{offset}} = \sum_{n=1}^{i} O_{f_{\omega i}}(n)P_{f_{\omega i}}(n)D(n)
\]

(7)

where \( D(n) = e^{-\alpha(n-i)} \) is the attenuation coefficient over time, \( \alpha \) is the attenuation factor. The smaller \( \alpha \) is, the smaller the selectable time range. The effect of \( \alpha \) on the AEC task will be analyzed in detail later in the experiment.

Based on the way of calculating the forward effective time range, the backward effective time range \( [i, i + b_{\text{offset}}] \) is also obtained, where \( b_{\text{offset}} \) is the backward corrected expectation value of offset. For the \( i \)-th frame, the final effective similarity time range is \( [i - \hat{f}_{\text{offset}}, i + b_{\text{offset}}] \).

After estimating the effective similarity time range of all frames, an effective similarity time range matrix \( \Phi \) is obtained. \( \Phi \) is composed of \(-\infty\) and 1, where similarity values outside the effective time range will be masked with \(-\infty\) following the settings in [11]. The cross-frame similarity matrix \( \Omega \) is truncated based on \( \Phi \) and normalized by row-wise softmax,

\[
EDC_{output} = softmax(\Omega \odot \Phi)X
\]

(8)

Figure 1: The calculation procedure of the proposed EDC.
3. Experiments and results

3.1. Dataset, Model, and Experiments Setup

To evaluate the performance of the proposed EDC, an acoustic event-related deterministic feature transformation method, the large-scale weakly labeled DCASE domestic environment acoustic event dataset [31] and the CHiME-Home [32] dataset for investigating the in-home sound source recognition are used in this paper for the AEC task. The audio clip in the DCASE dataset is 10-second, and the audio clip in the CHiME-Home is 4-second. The DCASE domestic audio dataset excerpted from Audioset [33] contains 10 classes of real-life polyphonic audio events. And 7 classes of different in-home audio events are annotated in the CHiME-Home dataset.

Convolutional recurrent neural networks (CRNN) perform well in AEC-related tasks [10, 29, 9], hence a typical CRNN in Figure 2 is used as the cornerstone model in this paper to test the performance of different data conditioning methods. In Figure 2, convolutional layers are applied to learn local shift-invariant patterns from features. To preserve the time resolution of the input, pooling is applied to the frequency axis only. Bidirectional gated recurrent units (BGRU) [9] are adopted to capture the temporal context information. The timemistributed dense layer has \( N \) units, where \( N \) is the number of acoustic event classes. When using the DCASE and CHiME datasets, \( N \) equals 10 and 7, respectively. In the last pooling layer, global max pooling (GMP) or global average pooling (GAP) is optional. The previous study [29] shows that GMP performs better than GAP in the AEC task. The reason may be that GMP encourages the response for a single location to be high, while GAP encourages all responses to be high [34]. An intuitive explanation based on the observation is the audio event does not appear in every frame in audio clips. As long as the event exists in certain frames, GMP can locate it and thus is more suitable for the AEC task. Therefore, GMP is used in the last pooling layer, from which clip-level probabilities of acoustic events can be obtained. Since EDC is inspired by self-attention, the trainable self-attention will be added after the convolutional layer of the model to compare with EDC. More details, please see the project homepage (https://github.com/Yuanbo2020/EDC).

In training, log mel-bank energy with 64 banks [35] is used as input acoustic features, which is extracted by STFT with Hamming window length of 40ms and overlap of 50% between the window. The loss function is binary cross-entropy, Adam optimizer [36] with a learning rate of 0.001 is used to minimize the loss. Under the same training conditions, the amount of training data for AM is twice that of OM. Therefore, to avoid the interference of different data amounts for model performance evaluation, early stopping is adopted in training. All systems are trained on a single graphical card (Tesla V100-SXM2-32GB) with a batch size of 64 for maximum 500 epochs. To avoid the influence of different threshold interference, the area under

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>frames</th>
<th>AUC (%) of OM</th>
<th>AUC (%) of AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>18</td>
<td>92.68</td>
<td>95.10</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>94.01</td>
<td>96.36</td>
</tr>
<tr>
<td>7</td>
<td>54</td>
<td>95.20</td>
<td>96.71</td>
</tr>
<tr>
<td>10</td>
<td>78</td>
<td>93.93</td>
<td>96.93</td>
</tr>
<tr>
<td>20</td>
<td>156</td>
<td>93.12</td>
<td>95.71</td>
</tr>
<tr>
<td>50</td>
<td>390</td>
<td>90.79</td>
<td>92.98</td>
</tr>
<tr>
<td>100</td>
<td>782</td>
<td>89.83</td>
<td>91.69</td>
</tr>
<tr>
<td>250</td>
<td>1956</td>
<td>88.11</td>
<td>89.16</td>
</tr>
<tr>
<td>500</td>
<td>3912</td>
<td>82.01</td>
<td>88.39</td>
</tr>
</tbody>
</table>

Figure 3: Visualization of frame-level representations distribution using unsupervised t-SNE [38]. Please note that models in this paper are trained by clip-level weak labels in datasets of DCASE and CHiME, and the label of each audio clip is a multi-hot vector, so the label corresponding to the frame-level representation is unknowable.

3.2. Results and Analysis

As shown in Equation 7, the attenuation factor \( \alpha \) affects the selectable time range. So the first research question (RQ) is:

• RQ1: What impact will different \( \alpha \) values have?

In experiments, raw audio clips are converted into frames with a hop size of 20ms between frames, resulting in the number of frames of input acoustic features of datasets DCASE and CHiME being 500 and 200, respectively. As shown in Table 1, \( \alpha = 7 \) and \( \alpha = 10 \) achieved best AEC results, the corresponding maximum selectable time range is 54 frames (1.08s) and 78 frames (1.56s). That is, for acoustic events in the two datasets, paying attention to information within 0.54s \( \sim \) 0.78s around their time position is the most effective. Previous studies [24] show the shortest duration of some daily audio events in real life is about 0.5s \( (0.29s \sim 0.53s) \). And events containing fluctuating or intermittent sounds usually consist of short events. Therefore, if the model can identify short-term acoustic events, it is able to recognize these events in a long term.

When \( \alpha = 500 \), the information within 1956 frames before and after the frame will be considered in its scope of attention. The number of frames of samples in the DCASE dataset is 500, so when \( \alpha = 500 \), the effect of attenuation coefficient is weakened, and EDC tends to focus on global information instead of local parts, the EDC cannot effectively focus on the event-related local information, so its performance is poor.

• RQ2: Compared with feature weighting, SpecAugment, Mixup and the trainable self-attention, how does EDC perform?

SpecAugment has three specific ways [25]: time warping, frequency masking, and time masking. The ablation study in Table 2 shows under the same training condition, the perfor
4. CONCLUSION

This paper proposes EDC, which is a deterministic feature transformation to gather event-related local information and enhance boundaries between events and backgrounds. The core of EDC is to adaptively select the frame-related attention range based on acoustic features to focus on the event-related local information. The experiments on two real-life polyphonic audio datasets of a total of 17 classes of events show that EDC performs well when compared with SpecAugment, Mixup, learnable feature weighting and self-attention in the AEC task. Future work will further analyze the impact of similarity probability distribution $P_{1,m}$ and attenuation coefficient $D(n)$ in the proposed EDC on diverse acoustic events in real life.

5. ACKNOWLEDGEMENTS

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

Table 3: Performance of different methods on the same model.

<table>
<thead>
<tr>
<th># Conditioning method</th>
<th>AUC (%) of OM</th>
<th>AUC (%) of AM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DCASE</td>
<td>CHIME</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>90.55</td>
</tr>
<tr>
<td>2</td>
<td>Self-attention [11]</td>
<td>84.36</td>
</tr>
<tr>
<td>3</td>
<td>SpecAugment [25]</td>
<td>92.61</td>
</tr>
<tr>
<td>4</td>
<td>Mixup [26]</td>
<td>93.44</td>
</tr>
<tr>
<td>5</td>
<td>Feature weighting [27]</td>
<td>94.22</td>
</tr>
<tr>
<td>6</td>
<td>Proposed EDC</td>
<td>95.20</td>
</tr>
</tbody>
</table>
6. References


