ICASSP2020: Duration robust weakly supervised sound event detection

Heinrich Dinkel and Kai Yu

Speech Lab, Shanghai Jiao Tong University, China

May 2020

Overview

Task description
- Problem statement
- Contribution

Experiments
- Data description
- Model description
- Evaluation protocol (metric)
- Experiment 1 - Median filtering
- Experiment 2 - Double Threshold
- Experiment 3 - Subsampling methods
- Experiment Samples

Conclusion

Task

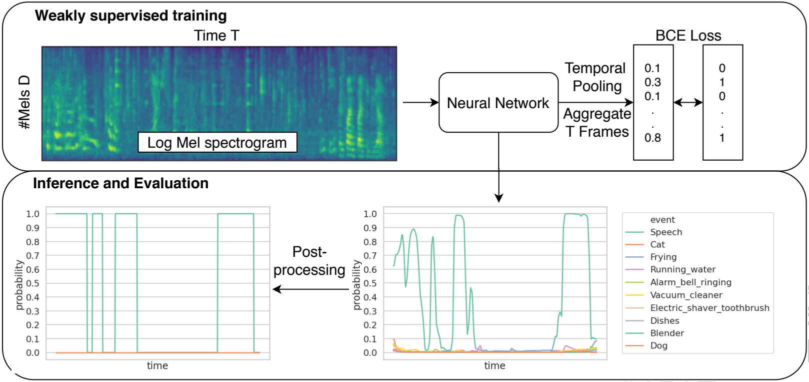


Figure: A typical weakly supervised SED framework.

Problems within weakly supervised SED

1. During training weak label estimates are obtained via mean/max-pooling the temporal dimension. This approach benefits long events and neglects short ones.
2. During inference, per-frame predictions are post-processed to smooth event predictions, using median filtering, which is shown to benefit long events further.
3. Neural network predictions are made on a fine scale (e.g., 20ms). Due to the noisy nature of this task, post-processing is necessary in order to obtain coherent predictions. However, post-processing cannot be learned by the network directly.

Contribution

1. Incorporate linear softmax as the default temporal pooling method
2. Using double threshold as a window-independent filtering alternative to median filtering
3. Subsampling the temporal resolution of our neural network predictions in order to learn event boundaries directly.

Development data distribution

This work utilizes the DCASE18 Task 4 dataset.

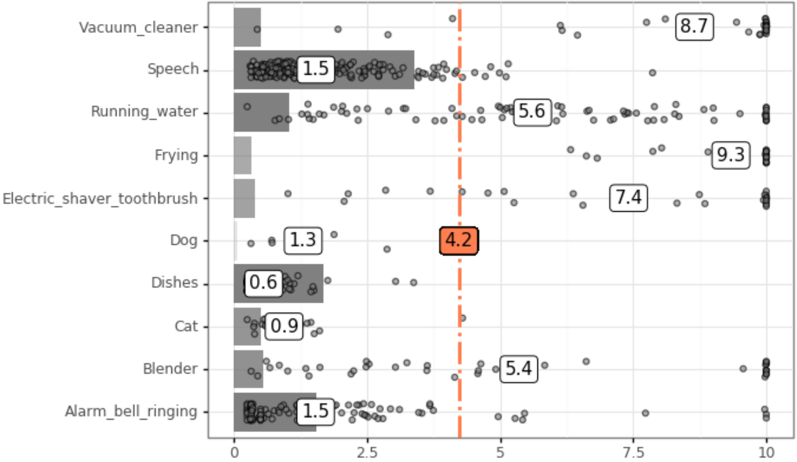


Figure: Duration distribution for the DCASE18 development dataset.

Development data distribution - Long and short events

We split the given development/evaluation data into two categories: short and long.
Short: Speech, Dog, Dishes, Cat, Alarm bell ringing
Long: Vaccum cleaner, Running Water, Frying, Electric Shaver/Toothbrush, Blender

Framework - Training

Only weak labels during training $\rightarrow$ temporal pooling. Let t be each utterance timestep and $y_t(c) \in [0, 1]$ its probability to output event c. We exclusively utilize linear softmax (LS) [1]:

$$y(c) = \frac{\sum_t^T y_t(c)^2}{\sum_t^T y_t(c)} \quad (1)$$

LS is a self-weighted average algorithm.

Framework - CRNN

Layer	Parameter
Block1	16 Channel, 3×3 Kernel
Subsample1	$s_1 \downarrow 2$
Block2	32 Channel, 3×3 Kernel
Subsample2	$s_2 \downarrow 2$
Block3	128 Channel, 3×3 Kernel
Subsample3	$s_3 \downarrow 2$
Block4	128 Channel, 3×3 Kernel
Subsample4	$s_4 \downarrow 2$
Block5	128 Channel, 3×3 Kernel
Dropout	30%
BiGRU	128 Units
Linear	10 Units
LS	

Table: CRNN architecture used in this work. One block refers to an initial batch normalization, then a convolution and lastly a ReLU activation.

Framework - Subsampling

We propose five temporal subsampling strategies $\mathcal{S}_k, k = 1, 2, 4, 8, 16$.

$$\mathcal{S}_1 = (1, 1, 1, 1) \tag{2}$$

$$\mathcal{S}_2 = (2, 1, 1, 1) \tag{3}$$

$$\mathcal{S}_4 = (2, 2, 1, 1) \tag{4}$$

$$\mathcal{S}_8 = (2, 2, 2, 1) \tag{5}$$

$$\mathcal{S}_{16} = (2, 2, 2, 2) \tag{6}$$

$\mathcal{S}_k$ represents subsampling by a factor of k.

Framework - Evaluation Metric

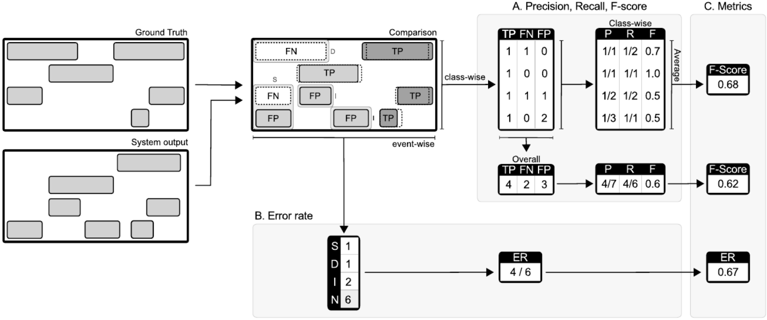


Figure: Event-level evaluation metrics. Taken from [2]

Framework - Basic Setup

- 2048 point, 40 ms Hann windowed log-Mel spectrogram every 20 ms (50 Hz).
- Evaluation metric: macro-averaged t-collar 200 ms, duration offset 20 %.
- Custom sampling strategy to tackle imbalance.
- Train/CV split: 90% / 10% with "balanced" events in each set.
- Optimization: Adam with learning rate scheduler.

Experiment 1 - Median filter influence

Here we utilize a standard CRNN model with $k = 1$. Median filter threshold $y_t(c) > \phi, \phi = 0.5$

	$\omega = 51$			$\omega = 1$		
Sub	Short	Long	Avg	Short	Long	Avg
Max	26.18	**23.78**	**24.98**	26.4	11.94	19.17
Mean	20.46	20.18	20.32	**28.26**	10.24	19.25

Table: Development F1-scores for different window size ω values of a median filter with respect to long and short clips.

Result: Median filtering improves performance average on cost of short events.

Experiment 2 - Double thresholding influence

$$\bar{y}_t(c) = \begin{cases} 1, & \text{if } y_t(c) > \phi_{hi} \\ 1, & \text{if } y_t(c) > \phi_{low} \\ & \text{and } y_t(c) \text{ in cluster where} \\ & y_t(c) > \phi_{hi} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

For all experiments: $\phi_{\text{low}} = 0.2, \phi_{\text{hi}} = 0.75$.

	$\omega = 51$			$\omega = 1$		
Sub	Short	Long	Avg	Short	Long	Avg
Max	16.82	**33.22**	25.02	31.36	28.52	29.94
Mean	17.28	31.48	25.29	**33.74**	27.88	**30.81**

Table: Comparison of double thresholding with different window sizes (ω) on the development set.

Subsampling methods

After these initial experiments we noticed that mean and max subsampling combinations can be benefitial to performance. Propose four new subsampling methods:

Name	Formulation
MM	$\text{mean}(x) + \max(x)$
α-MM	$\alpha \max(x) + (1 - \alpha)\,\text{mean}(x)$
LP	$\sqrt[p]{x^p}$
Conv	$\mathbf{W}x$

Table: Proposed subsampling layers. α is learned. p in L^p-norm is empirically set to 4.

Experiment 3 - Subsampling strategies

Configuration $\mathcal{S}_k$	1		2		4		8		16		Fusion $(2,4,8)$	
Subset	dev	eval	dev	eval	dev	eval	dev	eval	dev	eval	dev	eval
Winner-2018	-	-	-	-			-	-	-	-	25.90	32.40
Conv	27.26	14.97	23.04	19.95	32.05	22.46	24.80	21.13	16.39	17.07	25.26	23.68
LP	28.82	23.29	32.30	27.46	35.34	**30.81**	33.14	28.00	21.97	21.65	35.26	32.21
MM	30.35	24.72	35.64	29.80	27.98	25.14	31.15	28.20	20.11	21.83	36.29	31.01
α-MM	23.22	20.13	**36.00**	27.93	32.92	30.72	31.76	27.54	24.39	23.00	**36.44**	**32.52**

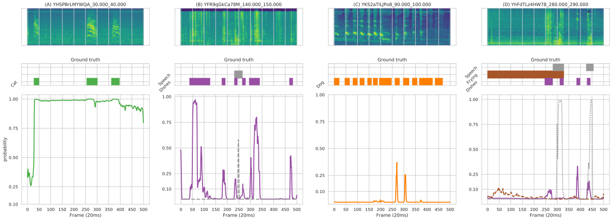
Table: Results for all four proposed subsampling types. Fusion is done by averaging the model outputs of $k = 2, 4, 8$. The Winner-2018 system is a fusion system. Results highlighted in bold are the best in class.

Experiment 3 - short and long events

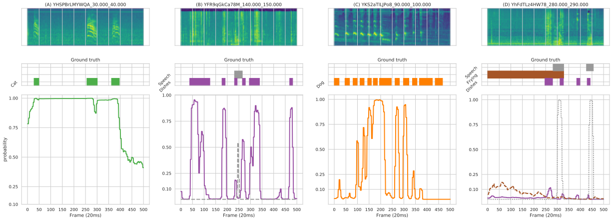
Type	Short	Long	Gap	Avg
Winner-2018	23.32	**40.36**	17.04	32.40
Conv	14.80	32.50	17.70	23.68
LP	**30.20**	34.22	**4.02**	32.21
MM	27.92	34.14	6.22	31.03
α-MM	29.66	35.40	5.74	**32.52**

Table: Short and Long clip results for evaluation data. Gap is the absolute difference between long and short clip F1 scores. All shown results are model fusions.

Experiment 3 - Visualize Short



(a) k=1



(b) k=4

Figure: Short samples for two LPPool models.

Conclusion

1. Median filtering improves Event-F1 performance on cost of short, sporadic events.
2. Double thresholding seems to be a robust alternative as post-processing.
3. Mixed mean and max subsampling methods improve performance.
4. LP seems to be the most stable subsampling function, while α-MM performs the best.
5. Our proposed adaptations improve event-level F1 performance both development and evaluation datasets.

Thanks

Thanks!
Code is available:
`https://github.com/RicherMans/Dcase2018_pooling`

- Y. Wang, J. Li, and F. Metze, “A Comparison of Five Multiple Instance Learning Pooling Functions for Sound Event Detection with Weak Labeling,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 31–35, oct 2019. [Online]. Available: http://arxiv.org/abs/1810.09050
- A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences (Switzerland)*, vol. 6, no. 6, p. 162, may 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162