Weakly-supervised deep learning approaches for sound event detection

Thomas Pellegrini, Léo Cances

GDR ISIS, 10-05-2019, Paris
Outline

Machine listening: motivation and problem statement

Multiple Instance Learning approaches

Limitations

Method enhancement
ANR JCJC Project (ANR-18-CE23-0005-01)

- Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning
- Start: 01 Oct. 2018
- End: 31 March 2022 (duration: 42 months)
- PhD student: Léo Cances, Post-doc to be hired
Machine listening

... extracting information from sound
Machine listening

... making sense of sound
Why weakly-supervised learning?

https://github.com/CrowdCurio/audio-annotator/

Why weakly-supervised learning?

Issues regarding data annotation

- What to annotate? Which granularity?
- Annotations are limited, expensive and time-consuming
- Annotations may introduce biases
- Annotations are noisy, "label noise"
- "Digital labor", les travailleurs du clic, Antonio Casilli
Sound Event Detection

Multi-label classification problem

- *Weak labels*: audio tags, $y \in \{0, 1\}^C$ \(\rightarrow\) many-to-one RNN
- *Strong labels*: frame-level tags, $Y \in \{0, 1\}^{T \times C}$ \(\rightarrow\) many-to-many RNN
We want to perform:

- "audio tagging": infer weak labels, $\hat{y}$
- "localization" or SED: infer strong labels, $\hat{Y}$

Input: a sequence of T.F. representation vectors, $X = \{x_1, \ldots, x_T\}$

Only weak labels are available for learning

In a recording weakly labeled as *Dog*, some acoustic frames will comprise dog barking and others will not
Weakly-labeled SED: approaches

Three main approaches:

1. Baseline brute-force approach: "False strong labeling"
2. Attention mechanisms: Gated Linear Units, for ex.
3. MIL-inspired approach: max MIL pooling function and variants
Multiple Instance Learning

- MIL terminology
  - $X = \{x_1, \ldots, x_T\}$: a bag or a set
  - $x_i$: an instance
  - No dependency nor ordering among instances
  - A single binary label $Y$
  - Unknown individual labels: $y = \{y_1, \ldots, y_T\}$

MIL assumptions:

\[
Y = \begin{cases} 
0, & \text{iff } \sum_t y_t = 0, \\
1, & \text{otherwise}
\end{cases}
\]

In a compact form:

\[
Y = \max_t \{y_t\}
\]
MIL: how to model the bag-level prob $P(Y = 1|X)$?

- MIL assumption: $P(Y = 1|X)$ must be permutation invariant since no ordering nor dependency between instances $x_t$
- Instances: "monomials"

**Theorem 1.** A scoring function for a set of instances $X$, $S(X) \in \mathbb{R}$, is a symmetric function (i.e., permutation-invariant to the elements in $X$), if and only if it can be decomposed in the following form:

$$S(X) = g\left( \sum_{x \in X} f(x) \right), \quad (3)$$

where $f$ and $g$ are suitable transformations.

**Theorem 2.** For any $\varepsilon > 0$, a Hausdorff continuous symmetric function $S(X) \in \mathbb{R}$ can be arbitrarily approximated by a function in the form $g\left( \max_{x \in X} f(x) \right)$, where max is the element-wise vector maximum operator and $f$ and $g$ are continuous functions, that is:

$$|S(X) - g\left( \max_{x \in X} f(x) \right) | < \varepsilon. \quad (4)$$
MIL: how to model $P(Y = 1|X)$?

Instance-level approach

- $f$: instance-level classifier that returns scores for each instance
- $\sigma$: aggregates the scores
- $g$: Identity function

Bag-level (embedding-level) approach

- $f$: maps instances to embeddings
- $\sigma$: maps the emb. to a single bag emb.
- $g$: bag-level classifier
- $\sigma$: "MIL pooling", basic non-learnable choices:
  - $z = \max_{t=1 \ldots T}\{h_t\}$
  - $z = \frac{1}{T} \sum_{t=1}^{T}\{h_t\}$
MIL loss example

\[ \text{loss}({X, Y_c}) = \text{binCE}(Y_c, \max_t \hat{y}_{tc}) \]

where

- \( Y_c \in \{0, 1\} \): ground truth label for class \( c \)
- \( \hat{y}_{tc} \in [0, 1]^T \): temporal predictions for class \( c \)
Example OK
Example KO

- Issue in distinguishing some classes
Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*, why??
Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying* → score = 0 for *Dishes*, why??

→ A caveat of the MIL pooling function (here max)
  - When a class most often co-occurs with another one
  - When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only
Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying* → score = 0 for *Dishes*, why??

→ A caveat of the MIL pooling function (here max)
  - When a class most often co-occurs with another one
  - When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

Solutions?
- Choose another MIL pooling function (ongoing work)?
- Data augmentation (ongoing work)?
- Penalize for prediction similarity
Adding a similarity penalty

\[
\text{loss} \left( \{X, Y_c \} \right) = \text{binCE} \left( Y_c, \max_t \hat{y}_{tc} \right) + \alpha Y_c \sum_{l \neq c} Y_l \max(0, \cos(\hat{y}_l, \hat{y}_c))
\]
Adding a similarity penalty
Prediction correlations

- Less confusions!
Speech
Dog
Results on DCASE 2018 Eval subset

<table>
<thead>
<tr>
<th>Approach</th>
<th>Official Baseline</th>
<th>GLU-MIL</th>
<th>GLU-MIL+cos</th>
<th>JiaKai</th>
<th>Liu</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-score (%)</td>
<td>10.80</td>
<td>22.60</td>
<td>26.20</td>
<td>32.4</td>
<td>29.9</td>
</tr>
<tr>
<td>Alarm / bell / ringing</td>
<td>4.8</td>
<td>29.0</td>
<td>30.4</td>
<td>49.9</td>
<td>46.0</td>
</tr>
<tr>
<td>Blender</td>
<td>12.7</td>
<td>23.3</td>
<td>27.7</td>
<td>38.2</td>
<td>27.1</td>
</tr>
<tr>
<td>Cat</td>
<td>2.9</td>
<td>28.7</td>
<td>30.3</td>
<td>3.6</td>
<td>20.3</td>
</tr>
<tr>
<td>Dishes</td>
<td>0.4</td>
<td>0.0</td>
<td>19.0</td>
<td>3.2</td>
<td>13.0</td>
</tr>
<tr>
<td>Dog</td>
<td>2.4</td>
<td>19.8</td>
<td>20.9</td>
<td>18.1</td>
<td>26.5</td>
</tr>
<tr>
<td>Electric shaver / toothbrush</td>
<td>20.0</td>
<td>6.2</td>
<td>19.1</td>
<td>48.7</td>
<td>37.6</td>
</tr>
<tr>
<td>Frying</td>
<td>24.5</td>
<td>27.6</td>
<td>21.2</td>
<td>35.4</td>
<td>10.9</td>
</tr>
<tr>
<td>Running water</td>
<td>10.1</td>
<td>13.3</td>
<td>13.2</td>
<td>31.2</td>
<td>23.9</td>
</tr>
<tr>
<td>Speech</td>
<td>0.1</td>
<td>37.6</td>
<td>35.0</td>
<td>46.8</td>
<td>43.1</td>
</tr>
<tr>
<td>Vacuum cleaner</td>
<td>30.2</td>
<td>40.6</td>
<td>45.2</td>
<td>48.3</td>
<td>50.0</td>
</tr>
</tbody>
</table>
Discussion on the cos penalty

Pros

▶ Increased the class-discriminative power of the networks

Open questions

▶ Larger variance
▶ Tends to force events to not overlap (but OK for the challenge)
▶ Better to penalize the prediction distributions? e.g. a Kullback-Leibler penalty?
▶ Worked for SED, generalization to other MIL problems?
Task 4 focus:

- do we really need real but partially and weakly annotated data or is using simulated data sufficient? or do we need both?
References

▶ T. Pellegrini, L. Cances, Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection, to appear in Proc. IJCNN, July 2019, Budapest, arxiv.org/abs/1901.03146