Project Final Report for MLSP: Few-shot Bioacoustic Event Detection

Kai Hu
kaihu@cs.cmu.edu

Xiang Li
xli7@andrew.cmu.edu

Yiming Pan
yimingpa@andrew.cmu.edu

1 Introduction

In many research fields, data labeling is costly both in time and resources. Although there might be some public datasets available, fixed ontologies or class labels are often a poor fit to a specific task. Few-shot learning (FSL) that aims to make predictions given limited labeled data can be a highly promising paradigm for such situations.

In this research, we apply FSL to bioacoustics, and propose a sound event detection (SED) system for animal (mammal and bird) vocalisations. The SED system will be expected to create a method that can extract information from five exemplar vocalisations (shots) of mammals or birds and detect and classify sounds in field recordings.

2 Related Work

Sound event detection (SED) is a machine learning task, which aims to automatically detect, label and localize sound-related events in a continuous audio signal. Due to the cost of manual labeling, only a limited number of training data can be accessed in many SED tasks, which necessitates the importance of few shot learning in this field.

Few-shot learning tasks have been increasingly studied in deep neural networks and often rely on meta-learning approaches including model-agnostic meta-learning[Finn et al., 2017] , Prototypical network[Snell et al., 2017], Relation network [Sung et al., 2018]. Most related works are in computer vision [Ravi and Larochelle, 2016] or natural language recognition[Yu et al., 2018]while little work has been done in audio-related tasks.[Shi et al., 2020] studies different few shot methods on acoustic event detection, and concludes that Prototypical network[Snell et al., 2017] is a good practice. We will not use deep neural networks directly in our research, but may borrow some ideas.

Several traditional approaches have been proposed to solve the few shot learning task for SED. Inspired by speeching recognition problems, we chose to implement Methods based on dictionary learning such as Non-negative Matrix Factorization[Cotton and Ellis, 2011] and Random Forest [Roma et al., 2013] as our baseline. Other models such as Support Vector Machines [Nogueira et al., 2013] and Kalman Filter [Podwinska et al., 2019] might be implemented in the future.

However, most of the models are primarily developed for speech processing and can't handle the few shot learning well. In this work we propose to combine such basic models and synthesize a new model that could complete both the SED task and few shot learning task.

3 Dataset

The dataset of DCASE2021 Challenge is pre-splitted into training and validation sets. The original form of dataset consists raw one-channel audio recordings combined with multi-class events annotations that explicitly separate the audio recordings into several segments and assigned one class or unknown class to each segment. The lengths for each segment can vary from several milliseconds to minutes.

In this project, we need to classify bioacoustic event in a few shot setting. We have a large training dataset with many labeled events on some seen classes. The inference is done on some unseen classes

Table 1: Dataset Summary

Models	Training Set	Validation Set
# of audio recordings	11	8
Total duration	14 hrs 20 mins	5 hrs
Total classes	19	4
Total events	4686	310

with several (1-5) positive examples. Audios are noisy and have different sampling rate. Sound events can have different durations. These heterogeneity might increase the difficulty for few shot setting. Some pre-processing could be necessary toward the task. In the following chapters, we have proposed several pre-processing methods and inference structures for the few shot setting, with comparison in later.

4 Methods

4.1 Problem Formula

We have a large training dataset $\{(x_i^{\text{Tr}}, y_i^{\text{Tr}})\}_{i=1}^N$ where $x_i^{\text{Tr}} \in \mathbb{R}^d$ is the input samples and $y_i^{\text{Tr}} \in \{0, 1, \cdots, M\}$ is the category label. During inference, we are given 5 samples $\{(x_i^{\text{In}}, y^{\text{In}})\}_{i=1}^5$ from a novel category, i.e., $y^{\text{In}} \notin \{0, 1, \cdots, M\}$. Then we need to do a binary classification on test samples to determine if they are from the same category as the given 5 samples.

4.1.1 few shot

Here we define the few shot setting as, the model can be created to make predictions from minimalist dataset. We want to predict the class of unseen task or fine-tuned in a very small sample. To be more specific for our project, during inference, we are asked to make predictions on some unseen class. We have no information about the class, but only 5 positive examples of this class. Then we get the test samples, and make a binary classification on whether theses test samples are in this class or not. Thus it is “few-shot”.

4.2 Baseline

For the novel category y^{In}, we are given 5 positive samples $\{(x_i^{\text{In}}, y^{\text{In}})\}_{i=1}^5$ and test samples z. We argue that z is more likely to be in the category y^{In} if z is closer to given positive samples.

For a distance function $d(x, z)$, we define the likeliness function $f(z)$. The probability that a test sample z is positive can be approximated by $f(z)$:

$$\text{Test sample } z \text{ is positive if } f(z) > \alpha \text{ for a threshold } \alpha$$

If we define the distance function as Euclidean distance $d(x, z) = (x - z)^\top (x - z)$, we can easily show that the above procedure is equivalent to compute the distance between the test sample and the center of given positive samples.

Before the prediction part, the input data is whitened with the sample variance. We define the whitening procedure and likeliness function as 1, 2, 3

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i^{\text{In}} \tag{1}$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x_i^{\text{In}} - \mu)(x_i^{\text{In}} - \mu)^T \tag{2}$$

$$f(z) = (z - \mu)^T \Sigma^{-1} (z - \mu) \tag{3}$$

In the baseline approach, the distance is compute directly from the input data, which might not be the best representation, since some dimensions of the input data may not be related to the classification problem. Include some irrelevant features might introduce more unwanted noise, some feature selection is important to improve the performance.

4.3 Canonical Correlation Analysis

One approach for the feature selection is Canonical Correlation Analysis (CCA). From CCA on the training data and labels, we can get a downsample projection on the data. Then we can use the pre-processed projection to select features.

4.4 Adaboost Feature

We train a boosting learning model on the training data. The model will find significant patterns on the training data for the classification task. We hypnosis that it can be transferred to the novel class and select features.
We use trained boosting model and take the test samples as input to get the the output probability vector. The system outputs a probability vector for any features. The k-th element of the vector represents the probability that the input is from the k-th category .We use the log of the probability vector as the selected feature in the inference stage. It can be the better representation for the feature selection. The log of the probability vector is better in the Euclidean distance.

4.5 Xgboost Features and Bayesian Corrected Whitening

In the baseline approach, we need to use the covariance matrix2 to do whitening for the input data. If we consider the few shot setting, the covariance matrix is too bias to be a good estimate and it's not invertible.
To overcome these problems, we proposed the following whitening method. First we have two assumptions.

Assumption 1 (mean distribution) *The distribution mean of different categories follows a Gaussian distribution. The prior distribution of the mean of the inference category is such mean distribution of the training data.*

Assumption 2 (covariances distribution) *Samples from different categories have different distribution mean, but similar covariance. Specifically, similar categories (distribution mean are close) have more similar covariance.*

Suppose we have M train categories with mean and variance. $(\mu_1, \Sigma_1), \ldots, (\mu_M, \Sigma_M)$. We define $\mu_0 = \frac{1}{M}\sum_{i=1}^{M}\mu_i$ and $\Sigma_0 = \frac{1}{N}\sum_{i=1}^{M}(\mu_i - \mu_0)(\mu_i - \mu_0)^T$. The priori distribution of category mean is $\mathcal{N}(\mu_0, \Sigma_0)$.
Let the sample mean and variance of the given inference positive samples be μ_{M+1} and Σ_{M+1}.
The posterior of the inference category variance is (d is some similarity function):

$$\hat{\Sigma} = \frac{1}{\sum_j d(\mu_j, \mu_{M+1})} \sum_{i=1}^{M+1} d(\mu_i, \mu_{M+1})\Sigma_i \tag{4}$$

The posterior of the inference category mean is

$$p(\mu | x_1^{\text{In}}, \ldots, x_N^{\text{In}}) \propto p(\mu)\Pi p(x_i^{\text{In}} | \mu) \tag{5}$$

With a MAP estimation

$$(N\Sigma_0 + \Sigma_{M+1})^{-1}(\Sigma_{M+1}\mu_0 + N\Sigma_0\mu_{M+1}) \tag{6}$$

4.6 Other Explored Methods

In this project, we have developed some drafts for several traditional methods. Although we didn't evaluate these methods in the final evaluation, we still listed them as references.

Table 2: Result Summary

Methods	AUC
Simply compute the similarity between given samples and test samples	52.8%
Compute the similarity on whitened data	54.2%
Compute the similarity on whitened CCA features	58.3%
Compute the similarity on whitened xgboost features	59.3%
Xgboost features and bayesian corrected whitening	61.1%

4.6.1 NMF

The original magnitude was decomposed into 50 non-negative bases. L1 regularization was added to ensure the sparsity of bases. The weights of decomposition are used as the input to the random forest classifier. To be noticed, although the input to the classifier is a list of segments and outputs are the labels for each segments, the nonnegative matrix was obtained by considering all the segments as a single matrix.

4.6.2 ICA

The sklearn version fastICA was used to separate the sources from audio. 20-dimensional source matrix was separated from the audio and is classified by the random forest classifier.

4.6.3 PCA

In one experiment, we performed PCA on the preprocessed data before feeding them into the random forest classifier. We kept 20 principle components. The result of using this approach is close to random guessing. The factor analysis (FA) version may be necessary.

4.6.4 Dictionary-based

In this experiment, we constructed a dictionary-based pipeline for this task. We applied a chain of filters on the 2D STFT magnitude and built a dictionary based on the concatenated output. The result is close to random guessing because we use gaussian filters on the spectrum and due to the characteristic of bioacoustics, these filters can't capture the pattern shown in the spectrum very well.

4.7 Evaluation

The evaluation was performed on validation set for all the methods. As the same procedure performed in the training set, only the time intervals with defined class labels were extracted and combined into a new audio recording. STFT with the same parameters was performed on the new audio recording and we evaluate the performance using the magnitude part from the frequency representation.

Given 5 positive samples, we compute the likeliness function for all test samples and rank the likeliness. With a given threshold, we predict all samples with a higher likeliness positive and the rest samples negative. We use AUC (area under the curve) of the true positive and false positive rate as our evaluation metrics.

Given a threshold, we can classify all test samples as positive or negative, and compute the false positive rate. With different thresholds, we will have different false positive rate. We use AUC to avoid choosing a threshold. The listening is not easy since the audio intervals are less than 1 second. For many samples, we did not even hear any sound. The training dataset is also filled with Unknown labeling and we have to filter them.

We repeated the experiments multiple times with different given positive samples and reported the average.

5 Discussion and Analysis

Unsupervised methods (PCA, ICA, NMF) are incapable for these problem settings. After we introduced the few shot settings, unsupervised methods couldn't produce meaningful output. These failure may due to some large noise in feature space, which means the greatest energy or variance may be the noise instead of some useful features. The unsupervised methods could not extract some useful features out of the noise, and they all failed to predict the result.

Supervised signals are important to build better performance in our setting, but whether the supervised method is strong or weak (linear CCA v.s. xgboost) does not matter much (the performance is not comparable to deep learning methods).

Using a priori to correct the estimations from a few samples is helpful, but the improvement is smaller than the first improvement: we assume the features are Gaussian distributions, which is not true.

Patterns and knowledge from different categories can still help. Our method significantly improved the baseline.

6 Division of work among team members

The workload is distributed equally among team members.

References

Courtenay V Cotton and Daniel PW Ellis. Spectral vs. spectro-temporal features for acoustic event detection. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 69–72. IEEE, 2011.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Waldo Nogueira, Gerard Roma, and Perfecto Herrera. Automatic event classification using front end single channel noise reduction, mfcc features and a support vector machine classifier. *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, pages 1–2, 2013.

Zuzanna Podwinska, Iwona Sobieraj, Bruno M Fazenda, William J Davies, and Mark D Plumbley. Acoustic event detection from weakly labeled data using auditory salience. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 41–45. IEEE, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Gerard Roma, Waldo Nogueira, Perfecto Herrera, and Roc de Boronat. Recurrence quantification analysis features for auditory scene classification. *IEEE AASP challenge on detection and classification of acoustic scenes and events*, 2, 2013.

Bowen Shi, Ming Sun, Krishna C Puvvada, Chieh-Chi Kao, Spyros Matsoukas, and Chao Wang. Few-shot acoustic event detection via meta learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80. IEEE, 2020.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018.