AUDIO CLASSIFICATION BASED ON WEAKLY LABELED DATA

Chieh-Feng Cheng, David V. Anderson, Mark A. Davenport[*] *Abbas Rashidi*[†]

Georgia Institute of Technology University of Utah

ABSTRACT

Audio event detection and classification are critical tasks in the analysis of multimedia data. Most current research on this topic focuses on processing strongly labeled data and using fully supervised machine learning techniques. However, many sources of multimedia data lack detailed annotation and rather have only high-level meta-data describing the main content of various long segments of the data. We propose a novel framework to perform audio classification when working with such weakly labeled data. A traditional approach to this problem is to use techniques for strongly labeled data and then to deal with the weak nature of the labels via post-processing. In contrast, our approach directly addresses the weakly labeled aspect of the data by classifying longer windows of data based on the clustering behavior of the acoustic features over time. We evaluate the proposed framework using both synthetic datasets and real data and demonstrate that our method can significantly outperform the traditional approach.

Index Terms— Audio classification, weakly labeled data, supervised learning

1. INTRODUCTION

This paper presents a novel method for audio classification based on weakly labeled training data. For weakly labeled data, sections of data are labeled as containing a signal of interest, but this signal may be intermittent and occur at one or more locations which are not clearly delineated. Weakly labeled data is common in many application areas, but is particularly common in audio classification tasks. For example, one might have training data consisting of clips of audio labeled “horns” that contain many other sounds along with some intermittent horn blasts. Given such weakly labeled data, our goal is to be able to classify segments of audio according to the content they contain, even if this content is only intermittent. Following the example above, we would like to be able to recognize when an audio segment contains a horn blast even if it only occupies a small fraction of the segment.

Part of the motivation for this research is in response to the deluge of self-recorded multimedia data now available. Many popular upload sites contain video and audio that lacks detailed annotation but rather only has high-level meta-data describing the significant content of the entire signal. Thus, given clip-level metadata, we only know that the described objects and events occur in the recording, but we have no information about how often and exactly when they occur.

More broadly, weakly labeled audio data arises in numerous other applications simply as a result of the difficulty and expense involved in manually annotating the precise contents of audio data. In this paper we consider one particular example application where this occurs in the context of acoustic monitoring of large construction sites. The goal in this application is to learn to identify the typical sounds of specific pieces of equipment and, where possible, their actions. Given weakly labeled training data for different pieces of equipment/actions, we would then like to be able to automatically monitor and characterize the activity at a construction site from simple audio recordings. Below, we evaluate our proposed framework in this context on real-world data we have collected from construction sites [1].

In this paper, we present a structured framework which combines multiple machine learning techniques as an approach to deal with these weakly labeled recordings. Our approach involves classifying longer segments of data by considering the clustering behavior of the acoustic features across a window of time to create a foreground / background model. On both synthetic and real-world data, we show that our proposed method achieves superior performance compared to a more traditional approach.

2. BACKGROUND

Audio classification has traditionally been studied using datasets which contain detailed temporal information of each sound event present [2, 3]. However, as noted above, many audio datasets are only weakly labeled in that the labels only indicate that some specific sound events are presented in the audio, but do not contain the exact time the events occur in the recording [4, 5]. However, the use of weakly-labeled audio data has received increased attention recently. Indeed, it was one of the subjects of the recent DCASE Workshop and Challenge—see, for example, [6, 7, 8]. In [9] the authors

[*] C-F. Cheng, D. V. Anderson, and M. A. Davenport were supported by NSF grant CMMI-1537261.

[†] A. Rashidi was supported by NSF grant CMMI-1606034.

use a fully connected neural network (FCN) to recognize instruments and tempo for each time frame of an audio clip with only the clip-level labels, extending this network to other sound event detection problems in [10]. In [11] the authors propose a multiple instance learning approach for sound event detection using weakly labeled data. Convolutional and recurrent neural networks (CNN and RNN) have also been used in the related context of audio tagging tasks [3, 12]. These existing works can be understood as first using weakly labeled data to build strong labels, and then applying standard machine learning techniques.

3. PROPOSED METHOD

In this paper, we advocate a new approach to training sound event detectors and classifiers on weakly labeled data based on jointly analyzing entire segments of weakly labeled data to create foreground / background models that implicitly learn the weakly-labeled events. Our proposed method for training on weakly labeled data comprises a sequence of interrelated steps. As an initial step, we note that it is typically helpful (but not required) to perform signal enhancement to reduce the background noise level. After the enhancement, the output data is then converted into a time-frequency representation using the *short-time Fourier transform (STFT)*. We then apply a dimensionality reduction technique to produce a low-dimensional set of features for each column (time bin/window) of the STFT – here we use a truncated *singular-value decomposition (SVD)*. We then apply K-means clustering to these feature vectors. From the output of K-means clustering, we can construct training vectors for segments corresponding to different categories by examining the distribution across the different clusters. A *support vector machine (SVM)* is then trained using these training vectors to identify different sound patterns (*e.g.*, corresponding to the various sound events of interest such as activities of each machine in jobsite recordings). Each of these steps are described in detail below.

3.1. Signal enhancement

We begin by simply noting that if there is a significant amount of noise, performance can be improved by applying basic noise suppression as a first step. Of course, the enhancement should be tuned carefully since low-level enhancement will still keep most of the background noise; while if the enhancement is too aggressive, the audio in the dataset might be distorted, degrading performance. For the construction site audio, we chose a classic signal enhancement algorithm developed by [13] because it has been proven to perform well in highly non-stationary noise environments such as what might be encountered at a construction jobsite. As shown in Figure 1, the frequency pattern is more distinct in the denoised recording than the original recording.

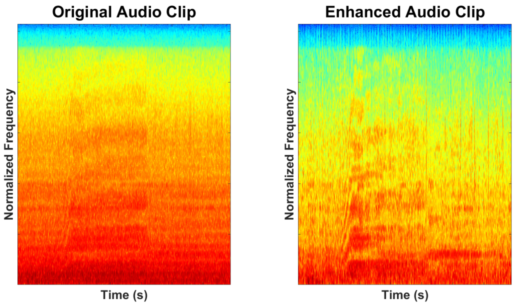


Fig. 1: Comparison between the STFTs of the original recording and denoised recording.

3.2. Feature extraction

The enhanced signal sampled at $F_s = 44\,100$ Hz is then converted to a magnitude time-frequency representation using STFT with a 512-point Hanning window, a 1024-point DFT, and a 50% overlap (256 overlapped samples).

In order to reduce the dimension of the STFT, we next compute the SVD of the (magnitude) of the STFT matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

By examining the matrix $\mathbf{\Sigma}$, which contains the singular values along the diagonal, we can determine how many components are sufficient to provide a good approximation to the original $\mathbf{X}$. We can then truncate the SVD by including only the first R columns of $\mathbf{U}$ and $\mathbf{V}$. We can then treat the columns of (the truncated) $\mathbf{V}^T$ as a low-dimensional set of features for each time bin of the STFT.

3.3. Clustering and forming the training data

We next apply K-means clustering to partition the columns of $\mathbf{V}^T$ into K clusters. This can be viewed as a way of characterizing the distribution of the columns of $\mathbf{V}^T$, The number of clusters is selected experimentally, and we found that six to eight clusters worked well for every case that we explored. However, we note that more complex signals may require more clusters. Our intuition is that different acoustic features in the signal will correspond to distinct clusters. If this is true, then as shown in Fig. 2, different categories will have some overlapping clusters—resulting from the common background elements shared by the different categories—and will have some non-overlapping clusters, which can be treated as representatives for each different category.

The K-means clustering results in each time bin being assigned a cluster label; but, this process is somewhat noisy and having data belong to a particular cluster is not necessarily a good class indicator. However, in practice what is often needed is a label associated with a slightly longer time period such as the duration of the sound or a short audio segment. For our data, the time period for a specific activity can last

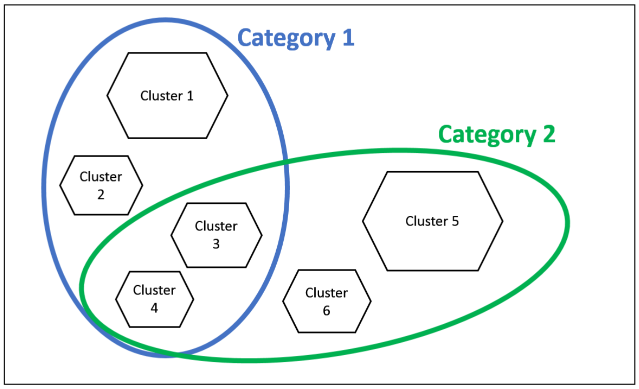
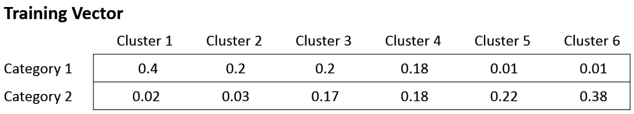


Fig. 2: Illustration depicting clustering behavior. The two categories share some clusters (corresponding to background features) but also contain clusters unique to each category.



Training Vector

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Category 1	0.4	0.2	0.2	0.18	0.01	0.01
Category 2	0.02	0.03	0.17	0.18	0.22	0.38

Fig. 3: Example training vectors for each category formed by computing an empirical histrogram across the clusters using a weakly labeled window of data.

for seconds, but each second will have hundreds of time bins. Thus, to construct training vectors for the full time period, we calculate empirical histograms which capture the distribution across the clusters within each time window. The length of the time window is set to be one second – long enough to capture a brief impact sound or a sustained sound according to the activities that we were trying to detect and classify. The one-second period was decided empirically by experimenting with several different audio datasets.

3.4. Classification via support vector machines

Following clustering, we then form a set of training data to be used by standard supervised learning techniques – in this case a support vector machine (SVM) [14, 15]. The input to the SVM is the normalized cluster membership histogram over the time-period of interest (1 second in our case). Two example training vectors are shown in Fig. 3. The two training vectors (rows) correspond to different events or categories in the audio clip and the columns capture the percentage of time bins that belonged to each cluster over a one-second window. We can repeat this process for many such windows for both categories of interest to form training data for each class, which can then be used to build a simple decision rule for classifying future data using SVMs. We use the *radial basis function (RBF)* kernel which was found to yield better performance than the linear kernel in our tests.

To train the SVM, we use the LIBSVM package in MATLAB [16]. To generate training data, we extract 30 to 40 seconds for each category from the dataset. The parameters C (trade-off parameter) and γ (bandwidth parameter) are selected by considering a log-scale range from 2^{-7} to 2^{6}. (Note that we select the parameters independently for each dataset.) We use 10-fold cross validation to select the appropriate values of C and γ. After training the SVM models for each machine, we extract other segments from the audio files (selected at random) as the testing data.

4. EXPERIMENTAL SETUP AND RESULTS

4.1. Experimental Setup

In order to evaluate the performance of the proposed system, we applied it to two different datasets. The first one is a synthetic dataset. The synthetic dataset consisted of audio spectrograms, generated so that each looked as though it contained multiple segments of sound each containing sound events from one of two classes interspersed with background sounds. To generate the synthetic dataset, a random sequence of states (representing weak labels) was generated and then for each labeled segment we further generate a random sequence of background and sound events consistent with the label. Each synthetically generated spectrogram column was generated randomly according to a predetermined distribution according to its type (background, event 1, and event 2). In different events, the spectral peaks of predetermined distribution varied considerably. The spectral peaks of predetermined distribution also varied in same event but they are generated in a small range. State 0 corresponded to environmental noise in the real recordings and so consisted of randomly generated Gaussian distributions with large standard deviation σ. For state 1 and state 2, both consisted of randomly generated Gaussian distributions with similar standard deviation but different mean μ of the distribution so that each state can represent different categories in real-life datasets. Finally, the synthetic spectrograms are blurred (convolution kernel [0.5 1 0.5]) to make the transitions between states less distinct and more realistic.

The second dataset consists of 8 different pieces of construction machines operating at various jobsites selected as case studies: 1) JD333E Compact Loader, 2) JD50D Compact Backhoe, 3) Ingersoll Rand Compactor, 4) CAT 320E Excavator, 5) Komatsu PC200 Excavator, 6) JD 700J Dozer, 7) Hitachi 50U Excavator, and 8) Concrete Mixer. Each machine was carefully monitored and the generated sounds while performing routine tasks were captured using a commercially available recorder (Tascam DR-05). Each audio file was manually labeled based on various activities that took place during the recording time. The label here will be used as correct label in the experimental results. Heavy construction equipment usually performs one major task (digging, loading, breaking, etc.) and one or more minor tasks (maneuvering, swinging, moving, etc.) in each cycle, so we classified each audio file

Table 1: Experimental Results

Machine	SVM Only		SVM with Filtering		NEW	
	Major Act	Minor Act	Major Act	Minor Act	Major Act	Minor Act
Synthetic Data	59.12%	60.39%	64.45%	67.69%	98.12%	99.0%
JD333E	69.12%	75.47%	82.12%	78.24%	80.03%	83.55%
JD50D	70.76%	56.78%	84.28%	59.87%	79.79%	76.74%
IR Compactor	68.45%	5.28%	80.55%	30.01%	82.47%	76.26%
CAT320E	64.67%	36.88%	78.24%	71.02%	80.36%	78.29%
Komatsu PC200	63.17%	57.41%	79.64%	69.81%	81.24%	77.48%
JD700J	69.98%	64.07%	81.33%	72.15%	80.06%	79.91%
Hitachi 50U	65.38%	52.93%	79.71%	54.96%	81.62%	78.88%
Concrete Mixer	62.49%	52.29%	77.82%	60.57%	80.16%	80.08%

based on two activities: major and minor (or activity 1 and activity 2). Also, within each activity time period, there will be some inactive times which only contain environmental noise in the recording. Thus, we will have audio clips each labeled with a specific activity, while these labels do not contain the information as what time the specific events occur in the clip. A large portion of the recordings might be environmental noise, which corresponds to state 0 in our synthetic dataset. For example, one recording for JD 700J Dozer could be manually labeled as "digging" from 0s to 30s but it might only dig for 10 seconds in this 30 seconds period. Each labeled audio file was sent through our audio processing pipeline and divided into activities 1 and 2. Finally, the performance of the algorithm for each case study has been compared to manually labeled files. The comparison results are depicted in Table 1.

The results of the proposed algorithm are compared with applying an SVM to the spectrogram directly and also to post-filtered SVM results—that is, choosing the activity based on a majority vote of the SVM results over the course of 1 second.

4.2. Experimental results

For the synthetic activity, we found that our method worked perfectly until we made the classes relatively similar, noisy, and blurred. Even under those circumstances, it performed far better than the filtered SVM method. It is obvious that the traditional SVM-based algorithm cannot identify different categories with weakly-labeled training data, while our new approach can complete the identification task well.

For the construction site recordings, as a more realistic dataset, we would generally expect the accuracy of all approaches to be lower than in the synthetic datasets. Nevertheless, our proposed method still performs relatively well and generally outperformed the filtered SVM methods (often significantly, e.g., see the Ingersoll Rand Compactor and CAT 320E Excavator). The overall accuracy of the proposed system was around 80% when identifying major activities and over 75% accuracy identifying minor activities for each machine. This is better overall than the filtered SVM. However, it is interesting that the filtered SVM methods performed better using the actual recordings than they did with the synthetic data. We believe that this is an artifact of how the data was collected. It is likely that the background sounds between recordings are somewhat correlated to the activity (for example, similar activities were recorded near the same time and place) so that the filtered SVM may be using the background not as a confuser but to actually help in the classification.

The main performance difference between our proposed framework and traditional filtered SVM algorithm involves identifying minor activities in our recordings. As shown in the table, the filtered SVM algorithm has difficulty when identifying minor activities in construction equipment recordings. The minor activities, which often contain significant environmental noise, inactive periods for a specific machine, and non-productive actions such as moving and swinging arms, result in overlapping clusters in our K-means clustering stage. The simple SVM approach cannot identify these activities well since they have a high probability of being confused with other target activities. This results, in practice, in overfitting to the background signal in these recordings. In contrast, in our proposed framework each different category will separate out the background into the "overlapping" clusters and the classification performance is determined more by the non-overlapping clusters, resulting in improved performance.

5. CONCLUSION

The proposed framework provides an alternative way to identify different categories given weakly labeled datasets. Our approach works by considering the clustering behavior of the acoustic features across a window of time to create a foreground / background model and automatically discount or ignore background or environmental signals. One strength of our approach is that we do not need a large database compared to existing neural network based methods – although one could easily imagine incorporating neural network architectures into our framework if desired (e.g., replacing the SVD with a restricted Boltzmann machine or similar autoencoder, and/or replacing the SVM with a neural network).

6. REFERENCES

[1] C.F. Cheng, A. Rashidi, M. Davenport, and D. Anderson, “Activity analysis of construction equipment using audio signals and support vector machines,” *Automation in Construction*, vol. 81, pp. 240–253, 2017.

[2] D. Ubskii and A. Pugachev, “Sound event detection in real-life audio,” *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events*, 2016.

[3] S. Adavanne, P. Pertilä, and T. Virtanen, “Sound event detection using spatial features and convolutional recurrent neural network,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017.

[4] Q. Kong, Y. Xu, W. Wang, and M. Plumbley, “A joint detection-classification model for audio tagging of weakly labeled data,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017.

[5] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. Plumbley, “Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging,” *arXiv:1703.06052*, 2017.

[6] S. Adavanne and T. Virtanen, “Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network,” in *Proc. IEEE Work. on the Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, Nov. 2017.

[7] J. Lee, J. Park, S. Kum, Y. Jeong, and J. Nam, “Combining multi-scale features using sample-level deep convolutional neural networks for weakly supervised sound event detection,” in *Proc. IEEE Work. on the Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, Nov. 2017.

[8] D. Lee, S. Lee, Y. Han, and K. Lee, “Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input,” in *Proc. IEEE Work. on the Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, Nov. 2017.

[9] J.Y. Liu and Y.H. Yang, “Event localization in music auto-tagging,” in *Proc. ACM Multimedia Conf.*, Amsterdam, The Netherlands, Oct. 2016.

[10] T.W. Su, J.Y. Liu, and Y.H. Yang, “Weakly-supervised audio event detection using event-specific Gaussian filters and fully convolutional networks,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017.

[11] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *Proc. ACM Multimedia Conf.*, Amsterdam, The Netherlands, Oct. 2016.

[12] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. Plumbley, “Convolutional gated recurrent neural network incorporating spatial features for audio tagging,” in *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, Anchorage, AK, May 2017.

[13] S. Rangachari and P. Loizou, “A noise-estimation algorithm for highly non-stationary environments,” *Speech Communication*, vol. 48, no. 2, pp. 220–231, 2006.

[14] A. Rashidi, M.H. Sigari, M. Maghiar, and D. Citrin, “An analogy between various machine-learning techniques for detecting construction materials in digital images,” *KSCE Journal of Civil Engineering*, vol. 20, no. 4, pp. 1178–1188, 2016.

[15] A. Ben-Hur and J. Weston, “A users guide to support vector machines,” *Data mining techniques for the life sciences*, pp. 223–239, 2010.

[16] C.C.Chang and C.J.Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.