CS 229 FINAL PROJECT
"A SOUNDHOUND FOR THE SOUNDS OF HOUNDS"
WEAKLY SUPERVISED MODELING OF ANIMAL SOUNDS
ROBERT COLCORD, ETHAN GELLER, MATTHEW HORTON

Abstract:
We propose a hybrid approach to generating acoustic models for recognizing specific species of animals based on their vocalizations using unsupervised analysis and recognition of similar sounds in a larger recording.

In many applications of machine learning to audio analysis and processing, such as voice recognition, the primary challenge is finding and annotating training data. This makes unsupervised clustering of similar sounds attractive as a means for creating acoustic models. The Cornell Lab of Ornithology hosts a large quantity of field recordings of a large variety of species of animals. However, many of these recordings are exceptionally long, and the animal in question sometimes will not be audible for several minutes at a time within these larger recordings. Searching and editing these long recordings to extract good audio data can be an extremely time-consuming process. However, analyzing audio on the basis of similar repetitive sounds could form a foundation for a potential solution to this problem. That is what this project explored.

Introduction:
The objective of this algorithm is to look for the *longest similar sounds* across a large recording. In previous work by Aren Jansen et al., these have been described as "pseudo-terms". To do these, we first need to find the following:

1) a suitable criteria for the similarity of two equal-length audio signals,
2) a robust method of detecting *consecutive* similarities over time,
3) a method to isolate these consecutive similarities and iterate, and
4) a satisfactory criteria for convergence.

Having solved these four problems, we can reliably say that the algorithm found the most common pseudo-term in a given recording. By searching for *most consecutive similarities* rather than *instances of highest similarity*, we can avoid matching short common ambient sounds (such as coughs and trees being hit), and assume that longer incidental sounds (forest noise, water, wind noise) resemble a Gaussian noise distribution.

Given a field recording with a significant amount of vocalizations by an animal, we aim to implement this method to find a pseudo-term that could be considered to be the most generic acoustic model for the vocalization of that animal.

Related Work:
Aren Jansen, during his time as a senior researcher at John Hopkins, originally proposed the method we used for finding our acoustic models as a way to comb through unannotated recordings of human speech. Unsupervised methods of clustering recordings of speech are attractive because of how expensive training data is for voice recognition software - a competent English language voice recognition solution on the scale of Apple's *Siri* or Google's *Google Now* will typically require around 80,000 hours of transcribed speech. Considering that it takes on average approximately 10 hours for a human to transcribe one hour of speech, it can be incredibly expensive and time-consuming to build reliable acoustic models for the human voice. This is why Aren Jansen proposed the approach to weakly supervised acoustic model training that we used for this project.

Dataset:
We used ten field recordings of animal sounds as our training dataset. The species represented in the recordings include chimpanzee, northern elephant seal, white-nosed coati, red deer, brown howler monkey, and red squirrel. Each of these field recordings lasts anywhere from twenty seconds to thirty minutes, and the represented animal vocalizes during each field recording several times. In all, this comprised fifty-five minutes and fifteen seconds of audio data. The field recordings were acquired with permission from the Macaulay Library at the Cornell Lab of Ornithology[2]. All processing and feature extraction of the data was done in Matlab.

Features:
The key feature of the audio data used in the generation of models is the Mel-frequency cepstrum (MFC) of the sound.The MFC is a tool used in audio analysis that represents short-term power spectrum of a sound. Mel-frequency cepstrum coefficients (MFCCs) are the individual amplitude values of an MFC. The process of finding the MFCCs at a certain point in time t is intuitive. First, we extract signal $x(t:t+w)$, wherein w is the window size. We then take that segment and get it's *smoothed cepstral values*:

$$X_{cep} = FFT(h \cdot IFFT(log(FFT(x))))$$

Here, we get the complex frequency samples of our signal, then take the log of the resulting spectrum. In doing this, we eliminate the phase component of the frequency samples, so when we run an inverse fourier transform on the resulting signal, all of the frequencies begin at $n = 0$. This allows us to smooth our cepstrum by truncating the signal. To truncate the signal, we multiply it by a window h, which we define as

$$h(n) = \{1,\ n < m;\ 0.5,\ n = m;\ 0,\ n > m\}$$

where $m \in [0, N]$ can be described as a smoothing factor for our cepstrum; the smaller the value of m, the less noisy our cepstrum will be. Next, we measure the energy of our cepstrum at specific points on the Mel scale. The mel scale is a logarithmic mapping of frequency using the following formula:

$$f_{mel} = 2595 \cdot log_{10}(1 + \frac{f_{hertz}}{700})$$

We chose the mel scale because by utilizing logarithmically spaced frequencies as our feature set, we get much more resolution at higher frequencies than lower frequencies. This means that our algorithm will be able to focus much more on the qualities of a sound that determine the *timbre* of an animal's vocalization, rather than the fundamental pitch of the sound (which can vary on a per-vocalization basis). An example of a Mel-frequency cepstragram, which can be described as the windowed MFCCs at fixed increments in time across the signal (similarly to a *spectrogram*), can be seen in *Figure 1*.

In order to further clean our data, we removed the DC component of the original field recordings, using a high-pass filter with a frequency cutoff as close as possible to 0 Hz, typically described as a *DC Blocking Filter.*

A hyperparameter which we had to rigorously test was the frequency band that we were generating MFCCs for. When analyzing voice, one can typically use a relatively narrow range of frequencies since the human voice does not produce any significant spectral content above 8000 Hz. However, for generalized animal vocalizations, we had to use a much broader range, since we want our algorithm to perform accurately with both low-pitched and high-pitched animal sounds.

Similarity Criteria:
When creating our similarity criteria, we wanted to make sure we were looking for spectral peaks, and avoid false negatives from variations in overall gain in the signal (i.e. if a dog is louder at 20:00 than he is at 25:00, we do not want to treat that as dissimilarity.) Therefore, we want to normalize our MFCCs on a per frame basis. Besides this, the similarity criteria we used is the same one proposed by Jansen:

$$K_{ij} = 0.5 \cdot (1 + (\frac{<x_i, x_j>}{||x_i, x_j||})^2)$$

Where x_i is our MFCC set at frame i, x_j is our MFCC set at frame j, and K_{ij} is the similarity of our file at timeframes i and j. Thus, K constitutes a *similarity matrix* of the recording with itself. Because of this, K is symmetric, and $diag(K) = 1$. Example similarity matrices can be seen in *Figures 2*.

A problem with this approach is that silence will result in very high similarities. The solution we ended up implementing was a simple noise gate- if there is no MFCC above a certain threshold at a certain point, we set its similarity to 0. This has the added benefit of greatly reducing the cycles our algorithm takes.

A Method for Finding Consecutive Similarities
After we compute our similarity matrix, we need to find consecutive instances of high similarity in respect to time. In the context of our similarity matrix, these will be line segments that run parallel to the diagonal, at $\theta = \frac{7\pi}{4}$. Again, we build on a solution that Jansen proposes: using a *Hough transform* to look for the highest density of similarity along any line running parallel to the diagonal. The resulting values will always increase relative to how close the line you are looking at is to the diagonal of the matrix. However, we can find local maxima in the resulting values to look for highest density of similarity at a specific time delta. From there, we go element wise through that diagonal to look for our longest line segment. A Hough transform is seen in *Figure 3.*

Removing Unique Sounds and Iterating
At this point, we take any instances of line segments and save the corresponding audio from the original file. After we concatenate those line segments together, we start over while linearly decreasing the hop size of our MFC cepstragram. This results in more time frames, thus giving us increased resolution in our cepstragram. A second-iteration similarity matrix is shown in *Figure 2.*

Checking for convergence
We stop iterating when we can no longer find local maxima above a certain threshold in our Hough transform. In the case that we find no peaks above our starting threshold, we decrease our threshold for local maxima.

Results:
We ran 10 different audio files through our algorithm. Of those 10 audio files, the algorithm did not converge for the entirety of one file (File 3) and another file was a duplicate (File 10). So we successfully analyzed 8 field recordings. In order to determine the efficacy of our methods, we decided that a successful output would be an output which contained a sound made by the animal that the original recording was meant to capture, measured by human perception. This seemed to be the most direct approach, considering the applications of these methods include cutting down hours of field recordings into a subset of short recordings with only the important audio.

However, for memory concerns, we broke these audio files into 10-second chunks as we processed them. Upon analysis of the data, this presented a problem. For certain animals, vocalizations can last almost as long as an audio chunk and can even span chunks. This does not lend itself to the analysis based on similarity and repetition that we propose in this paper. Also, often times, a chunk would only contain other noise (human speech, environmental noises, bumps on the recorder, the sounds of walking, etc.), and the output would be a subset of that noise.

For this reason, we considered two measurements of efficacy: the *cynical interpretation* and the *realistic interpretation*. The cynical interpretation measures success as the presence of an animal sound in an output. The realistic interpretation measures success as the presence of an animal sound in an output *only if* there was an animal sound present in the input. If there was no animal sound present in the input, that chunk is thrown out of the analysis.

This modification does not solve all of the problems with the data. For example, our algorithm is still unable to handle long, repeated vocalizations that might be better handled by larger chunks. However, it does solve many other issues.

Figure 4 shows the cynical interpretation versus the realistic interpretation in terms of percentages successful outputs per chunk in an audio file. Overall, the cynical interpretation gives our algorithm a success rate of 61.36%, and the realistic approach gives our algorithm a success rate of 81%.

Conclusions:

We believe the method described here for weakly supervised modeling of animal sounds shows much promise as a solution to the problem of annotating lengthy or otherwise unwieldy audio data. Based on preliminary results, it appears that the model is highly successful at identifying the longest similar sounds in a lengthy recording in order to find a generalized acoustic model for the vocalization of the animal species in the recording. More rigorous testing needs to be done on the model's outputs to determine its pitfalls and limitations and to further refine the algorithm. It is our belief that doing so is a worthy endeavor, as this model presents a unique solution to a common limitation in audio analysis research.

References:

A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training ," in *Proceedings of ICASSP,* 2013.

A. Norouzian, R. Rose, S.H. Ghalehjegh, and A. Jansen, "Zero Resource Graph-Based Confidence Estimation for Open Vocabulary Spoken Term Detection," in *Proceedings of ICASSP,* 2013.

Appendix A - Figures

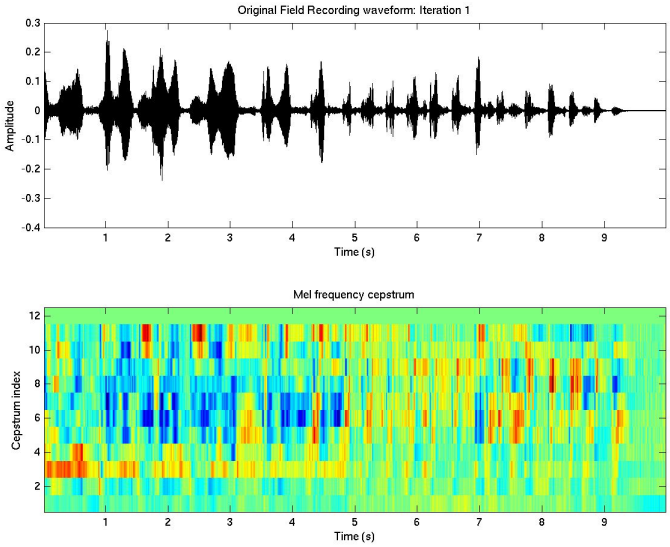

Figure 1: Matlab output showing original time-domain waveform and MFC of sound

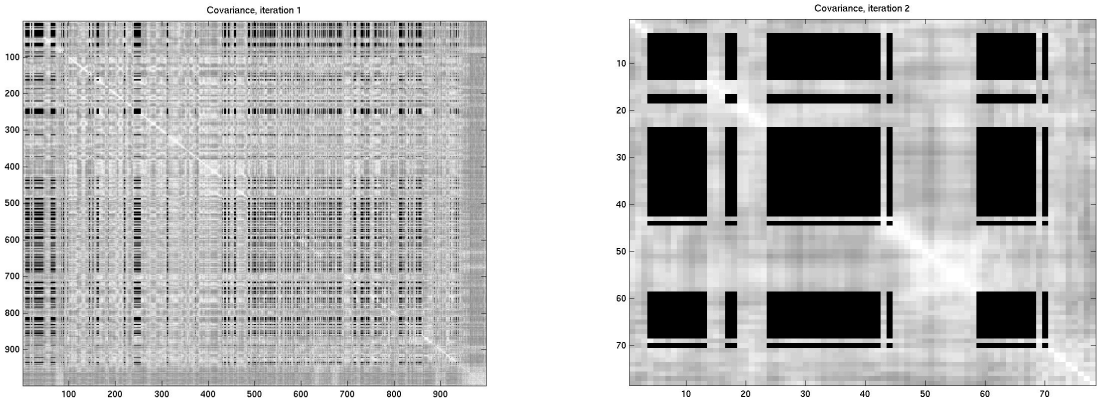

Figure 2: Similarity matrix of a sound segment with itself after one (left) and two(right) iterations

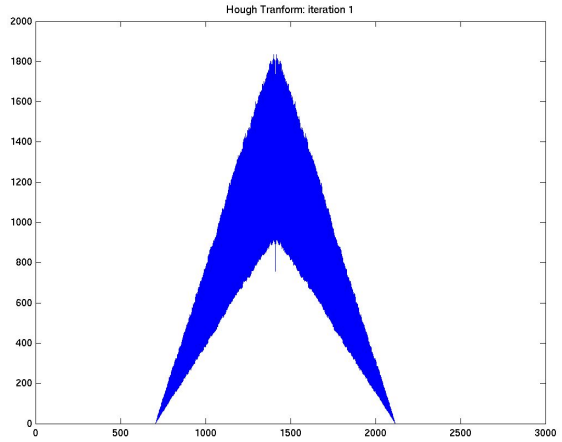

Figure 3: Hough transform of similarity matrix

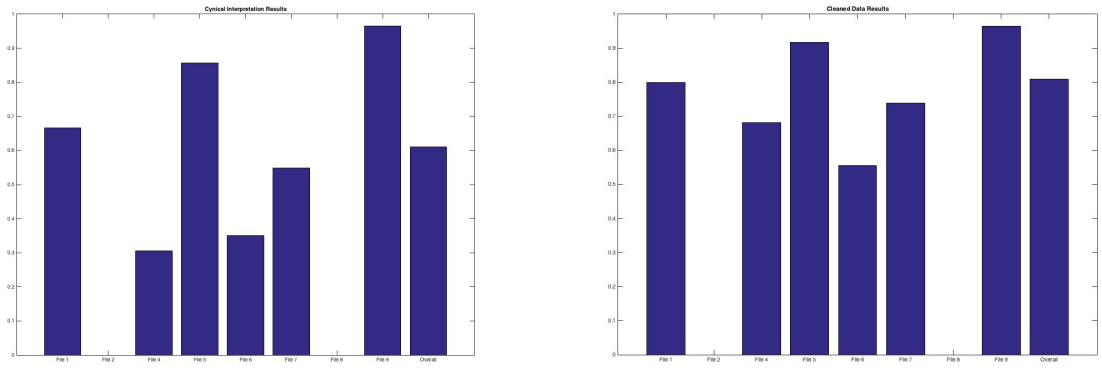

Figure 4: Success Percentages - Cynical Interpretation vs Realistic Interpretation