Identification of Avian Chirps and Songs of Forest Bird Species on Kauai Using Machine Learning - Milestone Report

Team Members: Ivy Wang (wangivy@stanford.edu)

Motivation
Use ML to create an algorithm that identifies which portions of an audio clip contains bird sounds.

Data Collection
I have reached out to Kauai Forest Bird Recovery Project and have obtained three hours worth of field recordings. The recordings are unlabeled and contain a lot of noise. I have manually labeled approximately some of this data and have preprocessed that audio to remove a lot of the noise. An example of the audio can be found at https://tinyurl.com/229a-bird-noise

Establishing Baseline With Audio Analysis
Initially, it appeared that the task could be easily done by applying a high pass filter to the data and then identifying sections with amplitude. This is equivalent of extracting features (has_high_frequency and has_high_amplitude) from sections of the audio. While this gave a very high recall rate, I realized that this method had extremely low precision, and would not have been a useful baseline. More features were needed to distinguish between true and false positives, which lead me to the use of Mel Frequency Cepstral Coefficients (MFCCs)[1].

Establishing Baseline With Previous Work
There are currently many algorithms that identify whether bird noises can be found within an audio clip that is a couple minutes long, as this was one of the tasks as this was one of the tasks for the DCASE 2018 challenge[2]. Many of these algorithms also make use of MFCCs.

I have attempted to establish a baseline using these algorithms by sectioning the audio clips and running each section through these algorithms to identify whether or not bird sounds appear. However, these do not generalize very well to identifying where the bird songs occur in the clip because the audio clip length required by these algorithms do not offer enough granularity for this task. Furthermore, these algorithms perform extremely poorly even for the tasks they were designed for when trained on the extremely noisy data in actual field recordings.

Method: Establishing Baseline With Logistic Regression
(I'm calling this a baseline, but there's so much to improve on this that it's not strictly a baseline… I elaborate on this further.)

In order to establish another baseline, I decided to implement a quick algorithm that uses logistic regression.

The algorithm works as follows:
1. Feature Extraction (Finding X): 26 MFCC ceptrum are first extracted from the data on frames of length 0.04 seconds with a step size of 0.01 seconds. These coefficients are then normalized and used as the features for the data.

---

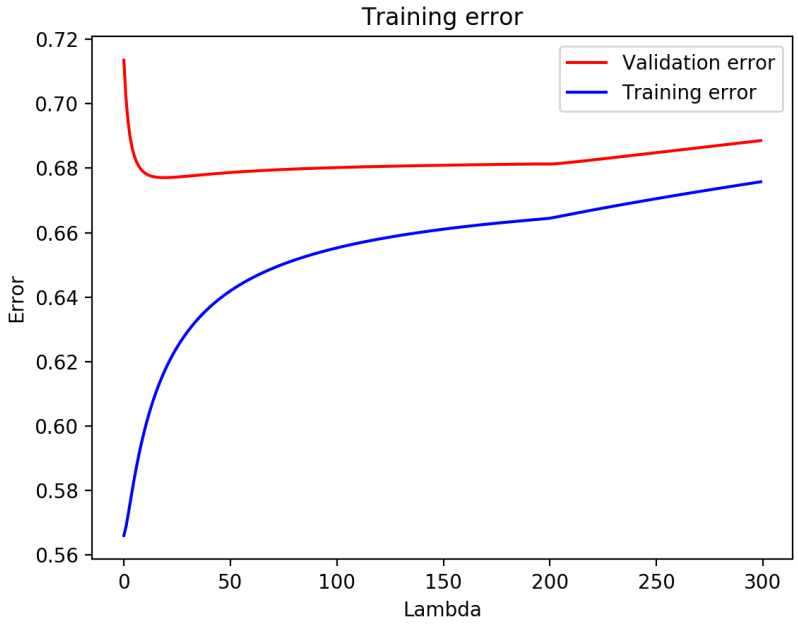
[1] https://github.com/jameslyons/python_speech_features
Thanks, Ferdinand, for the suggestion!

[2] http://dcase.community/challenge2018/task-bird-audio-detection-results

2. Label Extraction (Finding y): Each frame of data corresponds to a timestamp in the audio, so if the timestamp corresponds to a time interval that has been marked to contain bird sounds, then the label for that frame is marked as '1'. Otherwise, it is '0'
3. Finding θ: We perform gradient descent on the data to find the optimal θ that predicts the data (given regularization constant λ, and learning rate α)
4. Labeling new input: Once the training is done, we can obtain the feature vectors of a new audio clip, X_{test}. θ and X_{test} can be used to find the frames where the algorithm predicts that there is a bird noise. However, these are discrete positions. We transform these discrete positions into intervals by calculating the timestamps the beginning of the frame correspond to and smoothing over consecutive timeframes that are labeled '1'. Note that in this context, smoothing means ignoring intervals that are below a certain threshold in length.

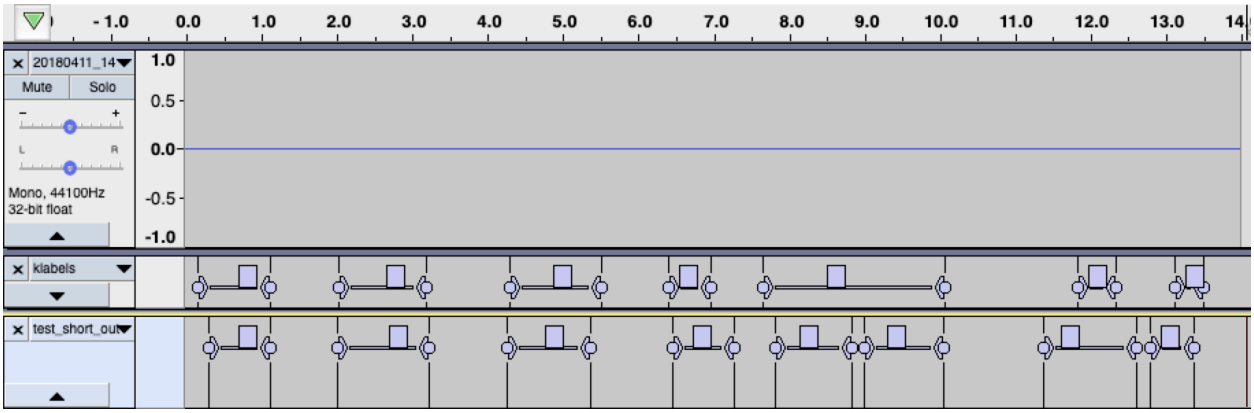
Preliminary Experiments (Analysis of current algorithm and how it helps with figuring out what the next steps should be)

We can see from the learning curves shown below that the error is quite high, and that there is still a gap between the training and validation errors, despite using regularization to prevent overfitting. These two observations allow us to determine what the next steps should be.



Since both errors are high, this suggests that logistic regression might not be able to fully capture the relationship between the coefficients and the labels. This is expected, as this implementation is our baseline. I hypothesize that a more complex algorithm such as a DNN would help address this.

However, one observation that gives doubt for the the above hypothesis is the performance of the algorithm when it is allowed to overfit. The results can be seen in the figure below. The uppermost horizontal bar shows the audio that is used to train the data on. The second horizontal bar shows labeled intervals corresponding to where bird noises are present. The third horizontal bar shows the output of the learning algorithm.



As can be seen, the output of the algorithm very closely matches the truth values, despite the feature space being significantly smaller than the number of points being labeled (14 features versus 1,400 inputs). This suggests that the model is complex enough to map the features used to the correct labels, which contradicts our hypothesis[3].

One explanation for this might be how the error is calculated. Currently, the inputs are continuous intervals, whereas the outputs of logistic regression are discrete frames that are identified to likely contain a bird sound. The output contain gaps that are mislabeled (due to bird noises having discontinuities, such as pauses between consecutive chirps), and the error calculation (used during gradient descent and shown in the first figure) penalizes these gaps. However, the algorithm then smooths the data, which gives the result that looks very successful.

One way to address this problem is to redefine the error function so that it compares the outputs once they are smoothed into continuous intervals against the continuous intervals of the inputs. However, this error function would not be differentiable, and so we would not be able to use gradient descent to solve this problem. To overcome this, we can use other algorithms to solve for the optimal theta using this new cost function (such as hill climbing with random restarts).

There is also a significant gap between the training and validation error, despite regularization. This suggests that the current algorithm might benefit from more training iterations. This makes sense, since the current algorithm is being trained on only 14 seconds of audio. The reason why the amount of training data is so small is because it is tedious to label the data. Furthermore, logistic regression is susceptible to bias in the data. Since the bird noises are so sparse in the field recordings (only a couple seconds of bird noises for every 5 minutes of audio), the algorithm obtains high accuracy (ratio of correct labels to total labels) by labelling everything as '0'. In order to balance the data, a large portion of the inputs that are labeled '0' are randomly chosen to be not used for the training data.

Concrete Next Steps
From the analysis above, it is worth spending the time to label more data for the models to train on. Furthermore, it will be worth exploring more complex learning models such as a DNN that can capture more a complex relationship between the features of the data.

---

[3] I'm not completely sure if this analysis is correct. I plan on asking about this during office hours.

However, the outputs of logistic regression still look very hopeful, and spending more time on modifying it so that (finding an alternative to gradient descent for the new cost function) it has more desirable behavior is also worthwhile.