Audio Segmentation of Kauai Forest Bird Vocalizations Using Logistic Regression

Ivy Wang
Dept. of Computer Science, Stanford University
Stanford, CA 94305
wangivy@stanford.edu

***Abstract*—The goal of this project is to use logistic regression to automate the task of segmenting audio from field recordings into relevant sections where bird vocalizations (calls and chirps) are present. This is an important pre-processing task in order to make further progress and use of existing machine learning algorithms designed to aid species identification. The results of this project show that logistic regression performs as well as state-of-the-art techniques for bird vocalization segmentation tasks, with the model capturing 88.9% of the bird vocalizations with a false positive rate of 8.6% on extremely noisy data, with the caveat of challenges that are presented with a limited amount of data. However, all current techniques remain bottle-necked by the quality and quantity of the data available to train these models.**

***Index Terms*—Audio segmentation, bird species identification**

I. Introduction

Even after 70 years of human point counts, basic demographics facts such as the population size of endangered forest bird species on Kauai is still unknown. Inaccessibility and cost considerations such as needing to fly into remote habitats make accurate numbers from human observations uneconomical. Without accurate numbers, conservationists do not know if conservation efforts are effective. Remote sensing and remote listening stations provide a fundamentally more economical approach to getting this data. However, despite the thousands of hours of data collected in these field recordings, very little progress has been made towards putting this data use due to the time-consuming nature of manually finding and identifying the species which are present.

The task of identifying a bird species from a pre-segmented, low-noise audio file has been a well-studied area in machine learning [3]. Conservationists from the Kauai Forest Bird Recovery Project (KFBRP) hope to apply these algorithms to field recordings in order to determine bird counts from field recordings. However, the important pre-processing step of segmenting the audio to isolate bird vocalizations has been under-studied and this tedious step is currently being done manually. This severely limits the research that can be done on both developing identification models on more noisy data, as well as the usefulness of such algorithms to conservationists.

The goal of this project is to aid the task of bird species identification by automating the pre-processing step of audio segmentation. Given the input of a .wav file recording obtained from KFBRP [1], the model uses logistic regression to predict the likelihood that vocalizations are present in a sliding window of small sections across the audio file. The algorithm them translates these predictions to timestamps corresponding to where the algorithm predicts bird vocalizations are present in the .wav file.

II. Related Work

Most previous attempts at performing this audio segmentation task has focused on using energy thresholds to identify the onset and offset of vocalizations. In these algorithms, audio is segmented into small windows of around 25 ms in length, and energy features are extracted from each window and compared. For example, Somervuo et al. apply an iterative threshold algorithm proposed in [2] which marks onsets and offsets using deviations from a background noise level estimate that surpass a specified threshold [4].

While extremely efficient and intuitive, one problem of this approach is that it assumes that a lower bound exists for the energy difference corresponding to when bird vocalizations are present. Furthermore, it assumes that all energy differences that surpass the threshold correspond to bird vocalizations. While these assumptions may be true in low-noise recordings, they fail to hold for field recordings which are extremely noisy. These incorrect assumptions give many false negatives and positives respectively when applied in a practical setting.

Another problem to this approach is the "border effect", which is a loss in accuracy caused by onsets and offsets which are non-instantaneous (like they would be in a practical setting) and span multiple windows. A solution proposed by Li, et al. to this problem is to "blur" the each window classification by pooling the classifications of nearby windows [2]. This gives a more smoothed result to which the threshold can be better applied.

The current state-of-the-art approach examines the audio file's spectrogram and identifies features in the spectrogram corresponding to bird vocalizations [3]. This is done by extracting features for each time-frequency unit in the spectrogram as a vector of frequency, the values in close to the time-frequency unit, as well as the variance of the values close to the time-frequency unit. These features are then used as the inputs to classify each time-frequency unit. The model is then trained by constructing a random forest classifier from the training data.

The use of time-frequency representations in this approach helps address the incorrect assumptions made by choosing energy difference features for classification. However, this approach is not as data-efficient, and generalizes poorly from the limited amount of training data available for our project.

III. Dataset

The data used for this project consisted of field recordings [1] in the form of 16kHz .wav files obtained from Kauai Forest Bird Recovery Project field recording database. The recordings were first preprocessed by running noise reduction in Audacity, then the ground truth was manually labeled using timestamps corresponding to the beginning and ending times of audio segments which contained bird vocalizations. However, the labeling process was imperfect due to human errors, so the resulting ground truth labels were imprecise and contained erroneous labels.

The positive labels in the data were extremely sparse, so the data was balanced by arbitrarily discarding sections of the audio where no vocalizations were present. This was an important step for preparing the dataset as only approximately 7.5% of the data contained bird vocalizations. This means that a logistic regression model that predicted negative for bird vocalizations on instances would have achieved approximately 92.5% accuracy. Balancing the data used ensured that this wouldn't be the case.

In order to obtain meaningful features from the field recordings, the audio file was split using a sliding window of length 25 ms and a step size of 10 ms. These window and step sizes were chosen so that each window contained enough information to capture an accurate representation of the audio data in the frame, but not so large that the variance across the audio window became meaningless. Mel-frequency cepstral coefficient (MFCC) features (using 26 cesptrum) were then extracted from each window to be used as inputs to our model. These features were chosen because the frequency bands used by MFCC approximates the response of the human auditory system. Furthermore, while MFCC features are a lossy representation of spectral data, it is the basis of many state-of-the-art audio recognition algorithms that are bounded by the limitations of human hearing [3] [5]. Other commonly used features such as energy were not chosen because the task required identifying bird sounds in a noisy environment where a bird vocalization did not necessarily correspond to such features.

The ground truth labels were also translated from timestamp intervals of where vocalizations were present. This was done by marking the windows whose timestamps corresponded to a positive interval as a positive sample, and the remaining windows as negative samples.

In total, the 30 minutes of imbalanced field recording audio data resulted in approximately 4.5 minutes of balanced data, which corresponds to an dataset size of 27055 samples (one for each window). This was randomly split in a proportion of 60/20/20 for training, validation, and test sets of sizes $16233, 5411, 5411$ samples, correspondingly.

IV. Methods (1-1.5 pages)

The model formulated for this task uses logistic regression to train and predict how likely each window of an audio recording contains bird vocalizations. Logistic regression is chosen because this is a supervised classification problem where high-quality data cannot be cheaply acquired. While more complex supervised classification models such as support vector machines and neural networks have also been considered, this lack of data would mean that sufficiently training these models will pose a significant challenge. One important thing to note is that while 27055 samples seems like a lot of data, many samples are repeated due to the repetitive nature of bird vocalizations, and the more complex models would have lead to high variance predictions. In fact, this is already the case, as we will further discuss in the next section.

The parameters of the model are expressed as a 1×26 vector θ. θ is applied to the 26 MFCC features to form a (scalar) linear combination of these features weighted by θ. This is then passed into the sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

to give a hypothesis for the likelihood. A vectorized equation for this prediction calculation of the i^{th} sample X_i (where X_i is a a 1×26 vector corresponding to the 26 features) is given by

$$h_\theta(X_i) = \sigma(X_i \theta^{\text{T}}) \tag{2}$$

θ is tuned using gradient descent. In each iteration, predictions are made for every single sample in our training set and θ is updated by comparing these predictions to the ground truth and minimizing the following cost function:

$$J(\theta) = \sum_{i=1}^{m} y_i \log(h_\theta(X_i)) + (1-y_i)\log(1-h_\theta(X_i)) + \frac{\lambda}{2m}|\theta|^2 \tag{3}$$

(where λ is the regularization constant) with the update formula:

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta) \tag{4}$$

(where α is the learning rate).

We perform the update described above for 2000 iterations and the resulting θ is then applied to obtain a hypothesis using equation (2). The predictions are then post-processed using the following protocol:

1) Smooth the predictions using the average value across a sliding window of 20 samples. This is done in order to combat the high variances across each sample.
2) Obtain the results by labeling each window as positive if the smoothed hypothesis is above a selected threshold c, and negative otherwise.
3) Smooth the results by using a majority vote protocol across a sliding window of 20 samples. This is done in order to address the "border effect" described in [2].

A plot of a subset of samples after each step is shown in Fig. 1 to give a better understanding of how the predictions are transformed in each step of the post processing.

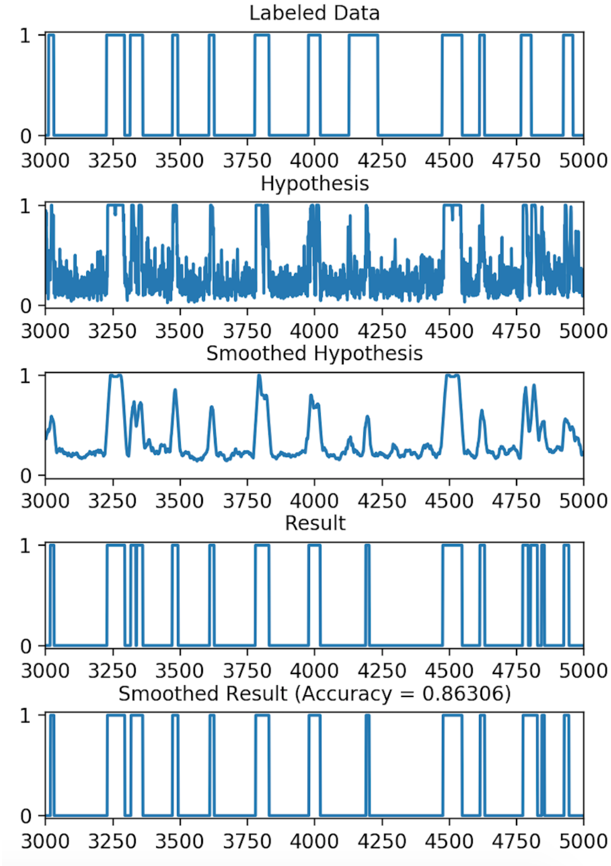

Fig. 1. Result of post-processing the predictions given by logistic regression. The horizontal axis represents the audio samples ordered temporally.The first plot is the ground truth, which is provided for the reader's convenience. Then in descending order, the plots correspond to the predictions, smoothed predictions, results, and smoothed results.

V. Experiments, Results, and Discussion

We run a series of experiments in order to select the hyperparamters used to train the model. These are described in the following subsections.

A. Tuning the number of training iterations and learning rate α

The number of iterations used and the learning rate α are tuned by plotting and examining the learning curve with respect to each iteration. As can be seen in Fig. 2, 2000 iterations with $\alpha = 0.3$ gives a learning curve with desirable convergent behavior.

B. Tuning the regularization constant λ

The regularization constant λ was tuned by comparing the accuracy (defined as the proportion of correctly labeled samples) of the trained model applied validation set and selecting the value that resulted in a minimal validation error. In this case, λ was chosen to be 0.1. However, one thing to note is that due to the limited amount of data available, the errors vary greatly depending on how the data is chosen for our training and validation sets. A way to reduce this variance and obtain a more accurate method for tuning the regularization constant would be to use cross validation error instead of validation error.

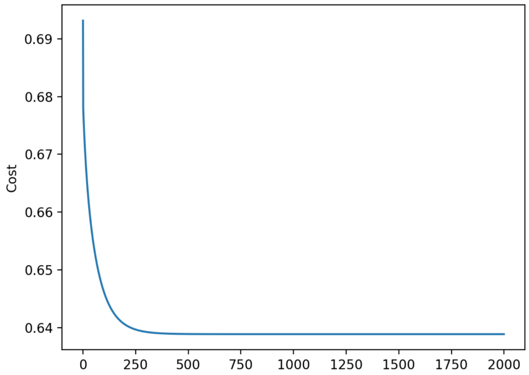

Fig. 2. Learning curve with $\alpha = 0.3$ across 2000 iterations of gradient descent. The asymptotic behavior at the number of iterations increase show that 2000 iterations is sufficient.

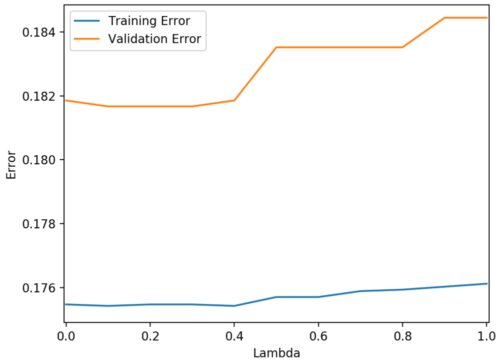

Fig. 3. Test and validation errors are plotted against each other to determine the best value of λ. Validation error is minimized at $\lambda = 0.1$, so this is chosen to be the regularization constant.

C. Results

To obtain the final results, the model is trained on the on the training set with the above hyperparameters applied. Fig. 4 and Fig. 5 show a sample of the resulting performance of the algorithm on our training and validation sets when a threshold of $c = 0.3$ is chosen.

After training our model, following confusion matrix is obtained for the model applied to the test set. A sample is defined to be correctly labeled if the smoothed result prediction obtained from the sample's input features matches the ground truth label corresponding to that sample.

One important thing to note is that the calculated accuracies are an underestimate of performance due to the imprecise nature of the ground truth labels. For example, bird vocalizations where there are multiple chirps are manually labeled (incorrectly) as a single segment of bird vocalizations, whereas the algorithm labels each chirp separately and does not mark

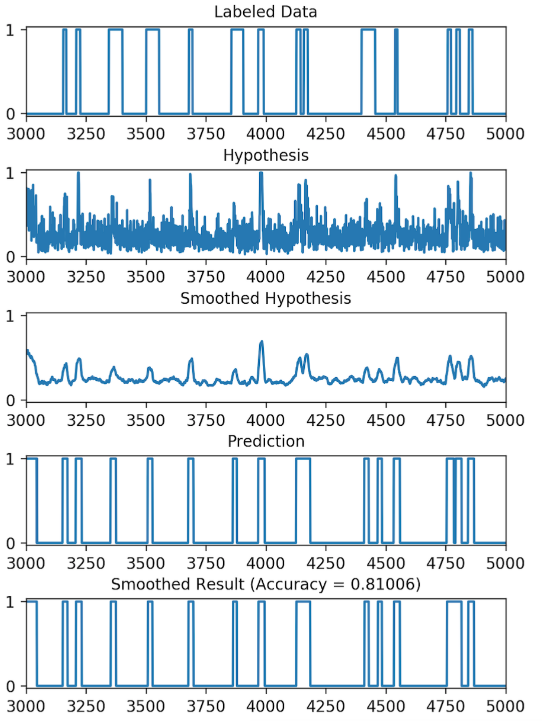

Fig. 4. Performance of trained model on training set.

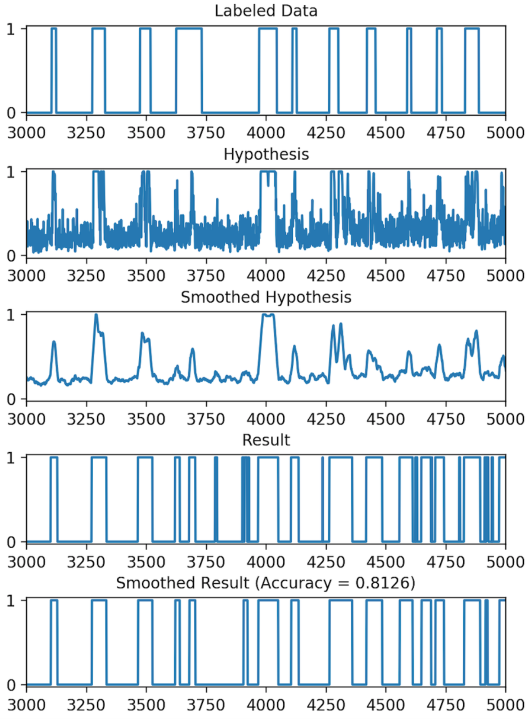

Fig. 5. Performance of trained model on validation set.

		Ground Truth	
		Positive	Negative
Prediction	Positive	TP = 2405	FP = 233
	Negative	FN = 300	TN = 2473

Fig. 6. This confusion matrix shows the results of the trained model applied to the test set.

the intervals between the chirps as positive. These intervals are considered false positives, despite the algorithm outperforming human precision. This suggests that the performance of algorithms on such tasks are currently being limited by the quality of data available, and more performant algorithms may be difficult to obtain without improving data quality first.

1) Accuracy Metric: The first metric used to understand the performance of the model is the overall accuracy of the results, defined earlier as the proportion of correctly labeled samples. Mathematically, this can be expressed as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Using this equation, we see that our data obtains an accuracy of 90.2% on our test data. This is surprisingly high, considering that it is higher than both the training and validation errors, which are 81.0% and 81.3% respectively. One possible explanation for this is that the limited amount of data available means that data in each set is not representative of the overall data. This also explains how the validation error is lower than the training error. The accuracy of our model on our test set may be anomalously high because the data corresponds to an easier classification problem.

2) AUC Metric: Another metric used to understand the performance of the model is examining the true positive rate (TPR) against our false positive rate (FPR) of our trained model on our test set. TPR and FPR are defined as follows:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{7}$$

We plot these values across varying values of the threshold c to compare the performance of our algorithm on our test set against the performance of the algorithms described in Section II to obtain the following ROC curves shown in Fig 7. Note that the ROC curves for Random Forest Classifiers was not obtained from our data, but rather the data obtained in [**?**], as not enough data was available to train such a model. However, the ROC curve is included as it is a useful baseline under the assumption that it is an upper bound to the performance of the model on our data. This assumption is made because the ROC curve given has been trained and tested on much larger set of less noisy data.

By examining the area under the curve, we see that our algorithm is on par with state-of-the-art techniques [**?**], and

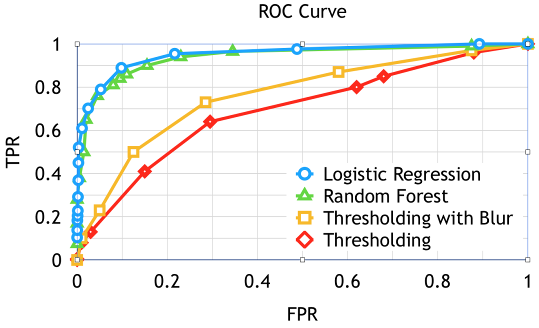

Fig. 7. ROC curves of our algorithm compared to other algorithms show that logistic regression on MFCC features performs as well as state-of-the-art techniques.

significantly better than frequency and energy thresholding techniques [4] used in the past.

However, one very large caveat to this is that our test data might not representative of our training data as evidenced by the anomalous accuracies, thus we cannot say much about the results unless

Finally, one interesting result to note is that the model remains performant with an accuracy of 83.0% when applied to the entire set of data, before it is balanced. This is shown in Fig. 8. This is expected, as logistic regression is only sensitive to imbalanced data during the training steps. However, this also suggests that there is potential for the model to generalize well on other sets of data.

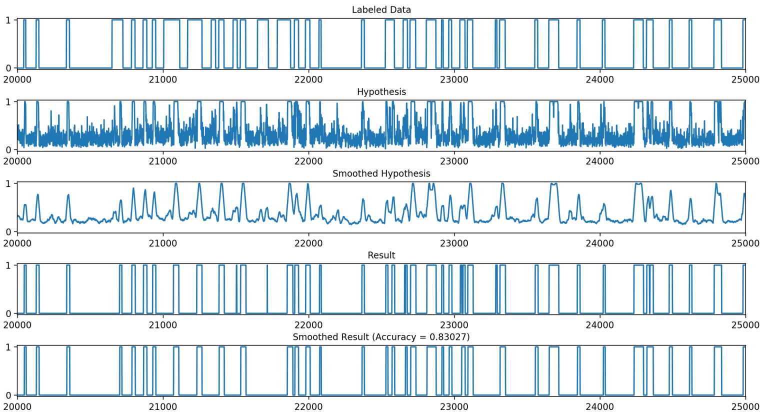

Fig. 8. The smoothed result of the trained model on an imbalanced data set suggests that the algorithm may generalize well to other sets of data.

VI. Conclusion and Future Work

With a threshold of $c = 0.3$, the model captured 88.9% of the bird vocalizations with a false positive rate of 8.6%, and appears to outperform the more commonly used audio segmentation algorithms in this area (with the aforementioned caveat). Given that the algorithms are currently bottlenecked by the quality and quantity of data labels, a necessary first step would be to obtain more labeled data of better quality.

Another consideration is that the current model only examines linear relationships between the MFCC features. One area to explore in the future would be to augment the feature space by including non-linear features, or even other supervised classification learning algorithms such as SVMs or neural nets. However, the above can only be done if more data is obtained.

Acknowledgment

Special thanks to Ferdinand Legros for the suggestion of using MFCC features and an improved accuracy metric, and Jonathan Dobbie for helping me obtain the KFBRP field recordings.

References

[1] *Kauai Forest Bird Recovery Project Field Recording Database* [Sound recording]. Available: https://tinyurl.com/229a-bird-noise.

[2] Li Dongge, et al. “Classification of general audio data for content-based retrieval.” *Pattern recognition letters* 22.5 (2001): 533-544.

[3] Neal Lawrence, et al. “Time-frequency segmentation of bird song in noisy acoustic environments.”2011 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011.

[4] Somervuo Panu, Harma Aki, and Fagerlund Seppo, “Parametric representations of bird sounds for automatic species recognition”, *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 2252 - 2263, November 2006.

[5] Xu Min, et al. “HMM-based audio keyword generation.”*Pacific-Rim Conference on Multimedia*. Springer, Berlin, Heidelberg, 2004.