Segmentation of Birdsong from Recordings of the H. J. Andrews Research Forest
DREU Student: Vivian Brown
Faculty Mentor: Xiaoli Fern
Summer 2010

1. Introduction

In order to analyze birdsong, it may be necessary to preprocess audio data by segmenting spectrograms (visual representations of short audio clips) into regions containing birdsong and silence. We compare two methods for generating bounding boxes around regions containing birdsong. The first method uses a threshold with hysteresis to identify louder regions within spectrograms. The second method uses a Random Forest trained on manually annotated spectrograms to assign a probability to each cell in a spectrogram.

For both methods, we identify a set of parameters that achieve a good balance between sensitivity and specificity by evaluating ROC curves over a range of parameter values. Using Random Forest, we achieve a per-cell true positive rate of .906 with a false positive rate of .043. Finally, to make it easier for human listeners to provide species labels, we apply the segmentation results to produce a clearer sounding audio recording with background noise suppressed.

This work will contribute to automatic species identification, which we expect will provide a more efficient way to collect data about bird population distributions. This will allow generation of species-based maps of bird activity and track changes with unprecedented temporal resolution, leading to a variety of applications in conservation and ecology.

2. Methods

2.1 Per-Rectangle Comparison

The first metric we used to measure the effectiveness of our segmentation was a per-rectangle comparison. For each rectangle in our generated “test” data set, we found the rectangle in the hand-annotated “true” data set that matched it most closely (judged by area of overlap between the two rectangles). We calculated the average overlapping area between test rectangles and their corresponding true rectangles. If a test rectangle did not overlap any true rectangle, the area of overlap was set to 0.

This approach was useful because it matched boxes in the test set with individual boxes from the true set. The boxes in the true set often marked distinct phrases and were annotated by species, so it was useful to seek a one-to-one match between boxes. The downside to this approach was that if a box in the test set overlapped two or more boxes from the true set, only the box from the true set with the largest amount of overlap was considered overlapping. The others were ignored.

The per-rectangle comparison was also problematic because it did not provide any information about the precision of our segmentation. It did not reflect the extent to which boxes in the true set were covered by boxes in the test set.

2.2 Rectangle Mask Comparison

The second metric we used was a comparison of the total area of overlap between bounding boxes we generated and bounding boxes in the true data set. To do this, we created two black and white masks, one for the test boxes and one for the true boxes. Areas within bounding boxes were set to white, and all other areas were set to black. Overlapping areas were areas colored white in both the test and true mask. These represented a true positive output. Areas colored black in both masks represented a true negative output. False positives were white in the test mask and black in the true mask, and false negatives were black in the test mask and white in the true mask.

Because the rectangle mask comparison considered every box in both the test and true set, it gave us information on both the accuracy and precision of our segmentation. It did not give us information about the degree of one-to-one correspondence between boxes in the true and test sets. We favored this metric over the per-rectangle comparison, leaving segmentation by syllable or species for a later step.

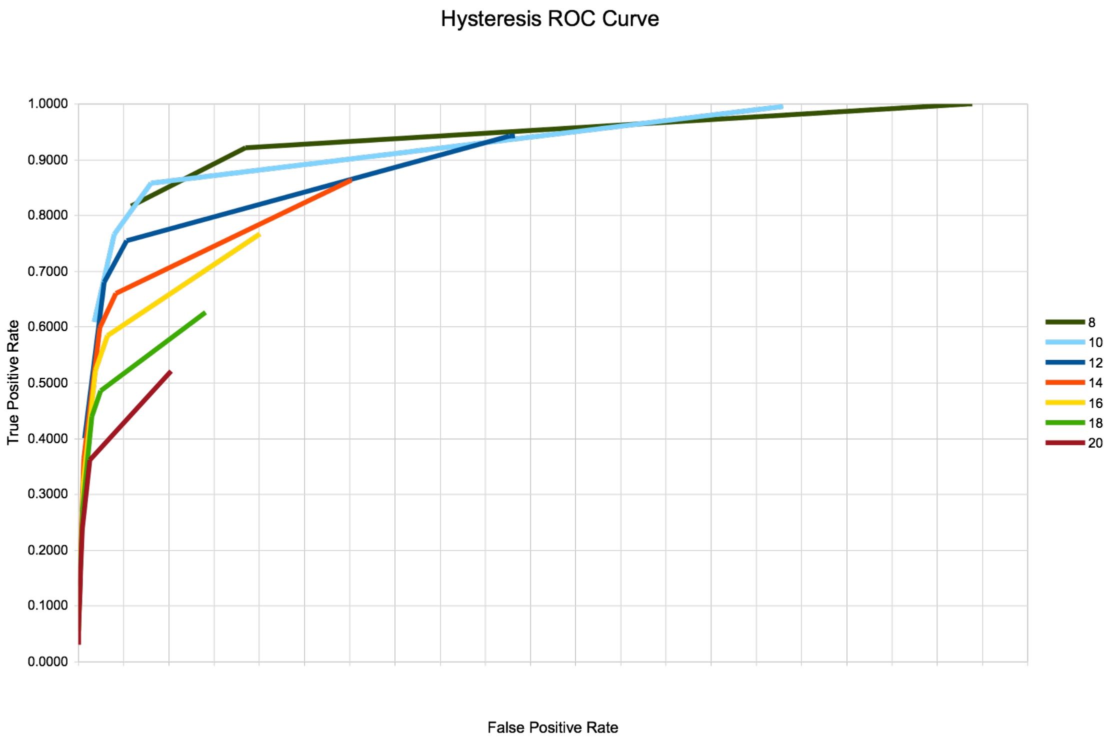
2.3 Hysteresis vs. Random Forest

We compared two general methods of segmentation using a rectangle mask comparison. The first, hysteresis, found “islands” of birdsong by searching for pixels with values over a given primary threshold, and then expending outward from those pixels using a depth-first search in order to include all surrounding pixels over a given secondary threshold. Bounding boxes were drawn around each island identified by hysteresis. The method second relied on a Random Forest to generate a probability mask describing the likelihood that each pixel within the spectrogram contained birdsong. The probably mask was blurred using a box blur of size eight. Hysteresis was then applied to the probability mask using the method described above in order to make bounding boxes.

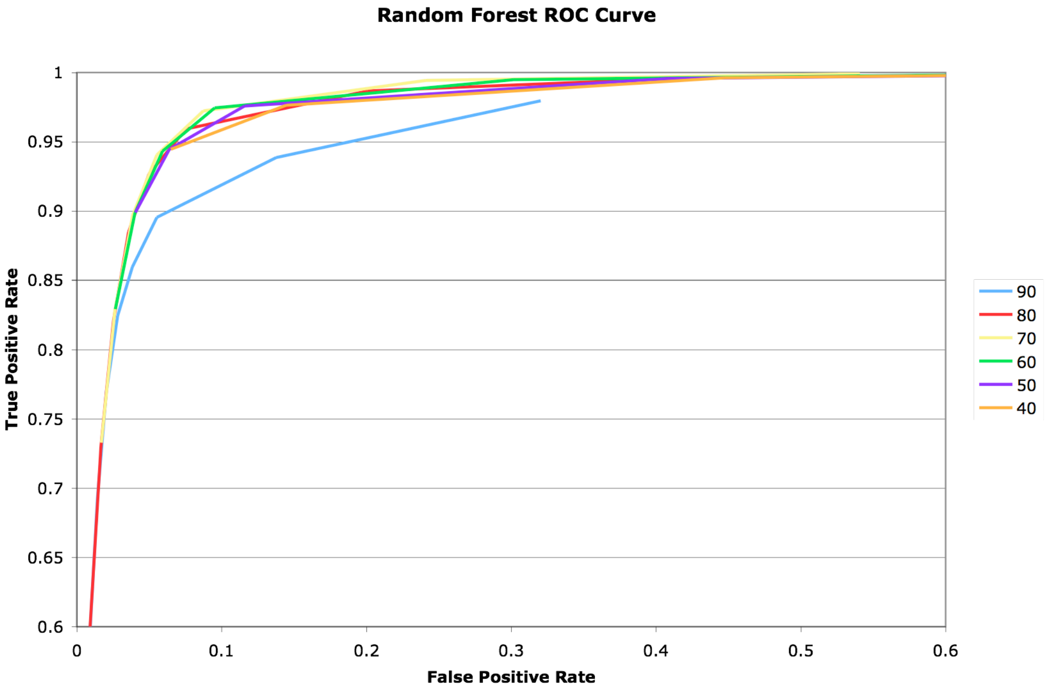
3. Results

We generated receiver operating characteristic (ROC) curves plotting the true positive rate against the false positive rate for each method at a variety of primary and secondary thresholds.

For hysteresis, we varied the primary threshold from .08 to .20 by multiples of .02. We varied the second threshold starting at .07 an increasing up to the primary threshold. We found that both primary and secondary threshold had a significant impact on true positive rate and false positive rate, and that hysteresis performed best when the primary and secondary threshold were equal. (Figure 1)



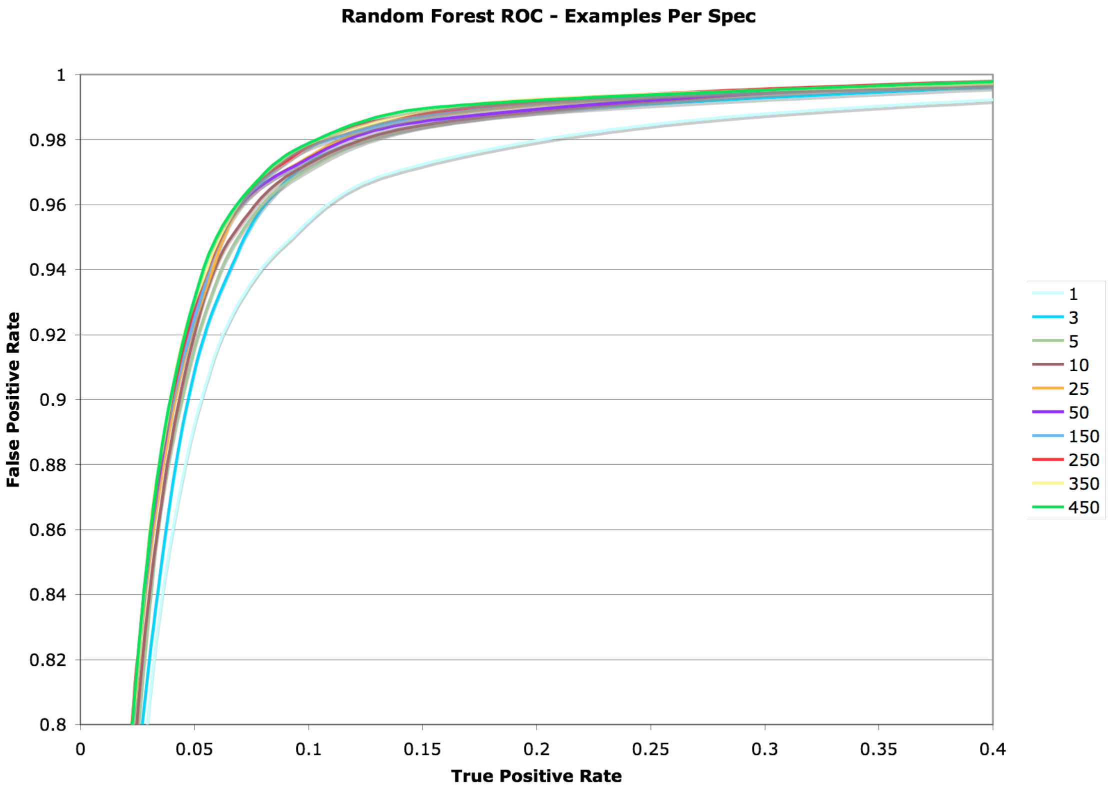
For random forest, we used 100 decision trees, a neighborhood size of 4, and 250 training examples per spectrogram. We varied the primary threshold from .4 to .9 by multiples of .1. We varied the second threshold starting at .1 and increasing up to the primary threshold. We found that varying the primary threshold had little effect on the accuracy of the segmentation. (Figure 2, note scale)



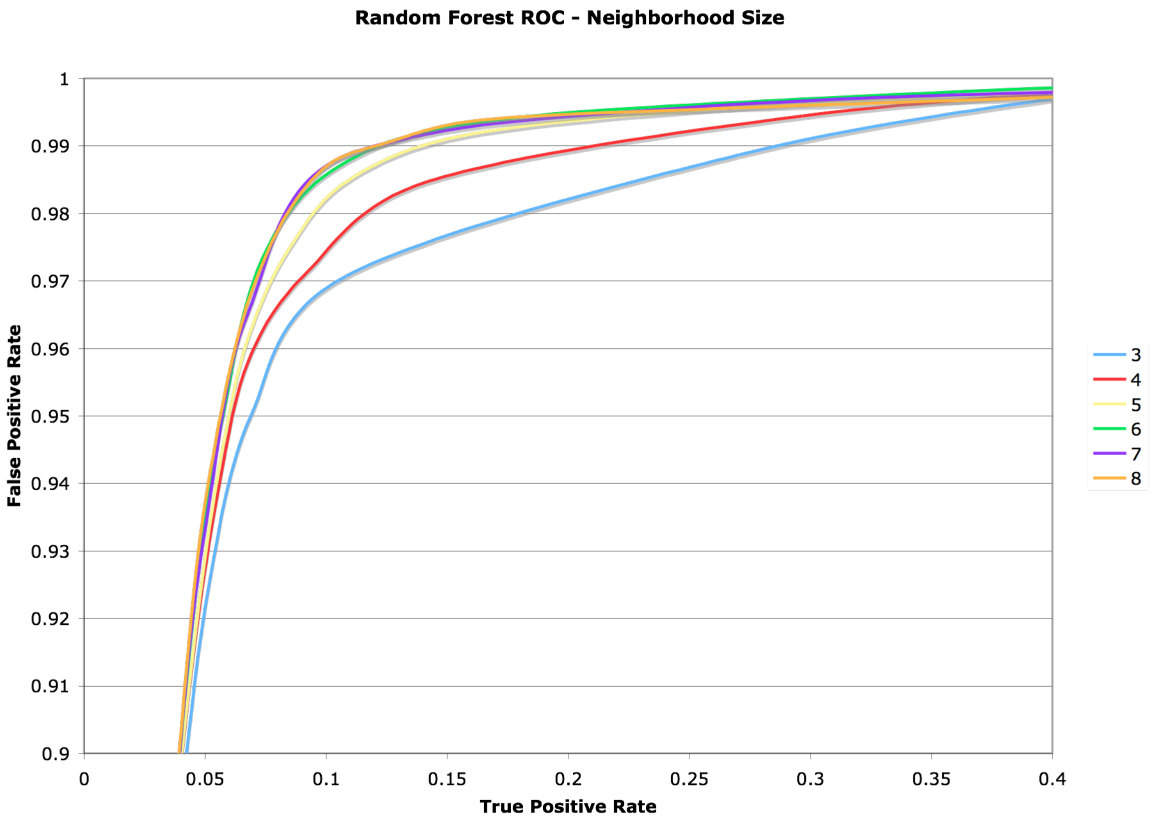
Comparing the output of hysteresis to random forest we found that hysteresis was significantly more accurate, and concluded that it was a better method for segmentation.

Next, we varied the parameters passed to random forest to find the best balance between speed an accuracy. Decreasing the number of decision trees used by random forest from 100 to 25 improved speed by a factor of about five with no loss of accuracy.

We varied the number of training examples extracted per spectrogram at a range of values from one to 450. We were able to reduce the training examples per spec to 25 without any significant drop in accuracy, using a neighborhood size of 4 and 25 trees. We noted that time and accuracy reflect the total number of training examples pulled from all spectrograms, and are therefore dependent on the number of spectrograms used for training. In this case, 11903 total training examples produced that best results. In the future, (Figure 3)



We varied the neighborhood size from three to eight. Neighborhood size denotes the radius of the of pixel values extracted to create a feature vector for a given pixel. Increasing neighborhood size caused a roughly linear increase in build and classification time. Accuracy continued to increase up to a neighborhood size of 6. (Figure 4)



Finally, at higher thresholds, random forest was identifying some small boxes containing no birdsong. We tried increasing the minimum box size identified by random forest from 50 pixels to a variety of values up to 1600. There was no significant gain in accuracy up to 200 pixels, and a decrease in accuracy for all values greater than 200 pixels.

4. Conclusion

In summary, we recommend using random forest for segmentation with a primary threshold of .7, 25 trees, 12,000 training examples, a neighborhood size of six, and a minimum box size of 2000. The ideal secondary threshold will depend on the desired balance between accuracy and precision, but will most likely fall between .4 and .65. A primary threshold of .7 and secondary threshold of .55 yielded a true positive rate of 0.908 and a false positive rate of 0.043.

Works Cited

[1] Breiman, Leo. “Random Forests,” Machine Learning. 2001; 45:5-32. Kluwer Academic Publishers (2001)

[2] Fagerlund. Seppo. “Automatic Recognition of Bird Species by their Sounds,” Helsinki University of Technology. 2004, Nov. 8

[3] Fagerlund, Seppo. "Bird species recognition using support vector machines," EURASIP Journal on Applied Signal Processing, pp. 64–64, January 2007.

[4] Harma, A. "Automatic identification of bird species based on sinusoidal modeling of syllables," in IEEE International Conference on Acoustics Speech and Signal Processing, April 2003, pp. V – 545–8 vol.5.

[5] Lakshminarayanan, Raich, and Fern, "Audio classification of bird species: a statistical manifold approach," in International Conference on Machine Learning and Applications, December 2009, pp. 53 – 59.