Ecological Informatics 21 (2014) 34–39

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Random Forest for improved analysis efficiency in passive acoustic monitoring

Jesse C. Ross [a,*], Paul E. Allen [b]

[a] Conservation Science Program, Cornell Lab of Ornithology, United States
[b] Cornell Lab of Ornithology, United States

ARTICLE INFO

Article history:
Received 21 June 2013
Received in revised form 6 November 2013
Accepted 5 December 2013
Available online 12 December 2013

Keywords:
Nocturnal flight call
Machine learning
Random Forest
Bioacoustics
Workflow

ABSTRACT

Passive acoustic monitoring often leads to large quantities of sound data which are burdensome to process, such that the availability and cost of expert human analysts can be a bottleneck and make ecosystem or landscape-scale projects infeasible. This manuscript presents a method for rapidly analyzing the results of band-limited energy detectors, which are commonly used for the detection of passerine nocturnal flight calls, but which typically are beset by high false positive rates. We first manually classify a subset of the detected events as signals of interest or false detections. From that subset, we build a Random Forest model to eliminate most of the remaining events as false detections without further human inspection. The overall reduction in the labor required to separate signals of interest from false detections can be 80% or more. Additionally, we present an R package, flightcallr, containing functions which can be used to implement this new workflow.

1. Introduction

There are few bioacoustic analysis processes that are completely automated and that perform well enough to avoid any human review or validation of the automatic outputs. Many acoustic monitoring studies suffer from this lack of automated processes, such that the scope and ambition of acoustic monitoring projects are often severely circumscribed by limited funding and availability of skilled analysts.

One important, yet understudied, arena of avian acoustic monitoring is the analysis of nocturnal flight calls (NFCs), the calls given by many North American birds during nocturnal migration (Evans and O'Brien, 2002). NFCs can provide a unique window into several aspects of bird behavior, life history, and migration, due to the fact that, unlike radar or thermographic information, NFCs can provide information about species identity of migrants (Farnsworth, 2005; Farnsworth et al., 2004; Gagnon et al., 2010).

Currently, simple band-limited-energy detectors are commonly used for passerine NFC analysis (e.g. Evans, 2012; Powers et al., 2013). These detectors can usually be tuned to produce an acceptable true positive rate, but often that tuning results in high false positive rates. It is frequently the case that 1% or fewer of events detected by these band-limited energy detectors are signals of interest (A. Klingensmith, personal communication). The human analysis effort required to sift through an overwhelming mass of false detections severely limits the scale of NFC analysis projects (Kunz et al., 2007).

In NFC studies, a typical analysis workflow would be:

1. run detector to produce candidate target signal events
2. manually review all detected events to sort events of interest from environmental noise or biological clutter
3. manually classify events of interest to classify into species or signal complex groups.

We present a modified analysis workflow which uses Random Forest, a machine learning algorithm, to eliminate many of the false positive events and reduce the overall labor required in a typical NFC analysis. Our approach uses existing, simple detectors. In short, we modify the typical workflow as follows:

1. run detector to produce candidate target signal events
2. compute spectro-temporal features for all events
3. randomly sample a small percentage of candidate events, and label these as signals of interest or false positives
4. use Random Forest technique to build a model that will use computed spectro-temporal features as predictors to classify events as signals of interest or false positives
   a. cross-validate model results
   b. increase labeled sample of events if model needs improvement
5. build a model trained on the sampled events, and apply it to the remainder of the events
6. discard the events labeled as false positives by the model
7. manually analyze events of interest to classify into species or signal complex groups.

The Random Forest algorithm (Breiman, 2001) is an ensemble classification technique which uses "bagging" (Breiman, 1996) to create multiple decision trees based on bootstrapped samples of a training

* Corresponding author at: Conservation Science Program, Cornell Lab of Ornithology, 159 Sapsucker Woods Road, Ithaca, NY 14850, United States.
E-mail address: jr634@cornell.edu (J.C. Ross).

1574-9541/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ecoinf.2013.12.002

set. A Random Forest model arrives at a classification by averaging the decisions arrived at by its component trees. An additional level of randomization is provided by creating each node of each tree based upon a random selection of the available classification features.

The Random Forest algorithm is well suited to this type of analysis, for at least four reasons (Liaw and Wiener, 2002). First, Random Forest does not require extensive tuning of parameters, tending rather to perform well on its default settings. Second, Random Forest performs well in the prediction of polymorphic categories, such as the wide variety of flight calls expected from many species of interest, and the many different types of false detections from band-limited energy detectors. Third, Random Forest is non-parametric, does not require transformations of predictors, and does not require predictors to be independent. Finally, Random Forest is tolerant of noisy or even meaningless predictors. This allows researchers to generate a large set of predictors for each event with minimal concern for the effects on model performance of any meaningless or correlated predictors in our predictor set.

Random Forest classifiers have been used successfully in species classification of calls of birds (Briggs et al., 2009; Keen et al., this issue), bats (Armitage and Ober, 2010), and dolphins (Barkley et al., 2011; Henderson et al., 2011). However, to our knowledge they have not yet been used in the classification of broad, polymorphic categories such as the vocalizations of multiple target species versus the many possible types of false detections.

We demonstrate that it is possible, using Random Forest, to considerably reduce the human effort involved with the typical NFC analysis workflow.

Finally, we make available an open source R package, flightcallr (Ross, 2013), containing the code used to implement this improved workflow.

2. Materials and methods

2.1. Data collection

The sound data we used were recorded by the Conservation Science Department at the Cornell Lab of Ornithology, at six locations in New York state (Powers et al., 2013) over 17 nights in September 2012. The recording locations were in Allegany, Lewis, Madison, Ontario, Tompkins, and Wyoming counties. A total of 1301 h of sound data was recorded. All recordings were made using Wildlife Acoustics Song Meter 2 recording units and SMX-NFC microphones. The sounds were recorded as 16-bit, 24 kHz uncompressed WAV files, since the acoustic energy of all known flight calls is concentrated under 12 kHz (Evans and O'Brien, 2002).

2.2. Initial detector run

We used Raven Pro 1.4 (Charif et al., 2004) to run two band-limited energy detectors over the entire sound stream from each recorder.

Table 1
The parameters used for band-limited energy detectors in Raven Pro.

Parameter	High-band detector value	Low-band detector value
Minimum frequency (Hz)	6000	2250
Maximum frequency (Hz)	11,000	2750
Minimum duration (ms)	27.2	30.8
Maximum duration (ms)	400	329.3
Minimum separation (ms)	101.6	49.0
SNR minimum occupancy (%)	25.0	20.0
SNR threshold (dB)	3.5	4.0
Noise power estimation block size (ms)	4997.7	999.5
Noise power estimation hop size (ms)	246.7	249.4
Noise power estimation percentile	50.0	50.0

Full parameters for each detector are listed in Table 1. The low-band detector ran in the frequency band 2250–3750 Hz, detecting vocalizations of larger-bodied migrants such as members of the family Turdidae (Evans and O'Brien, 2002). This detector produced a total of 1,320,634 events. The high-band detector ran in the frequency band 6000–11,000 Hz, detecting vocalizations of smaller-bodied migrants such as members of the family Parulidae (Evans and O'Brien, 2002). This detector produced a total of 465,770 events.

Our a priori expectations of species present were derived from knowledge of status, distribution, and migration phenology which is widely shared among birders, as well as knowledge of which birds vocalize in night flight (Evans and O'Brien, 2002). The eBird citizen science project (eBird, 2012) is another good source of data on status, distribution, and migration phenology.

2.3. Computation of spectro-temporal features for events

We used the R sound analysis package seewave version 1.6.5 (Sueur et al., 2008) in R version 2.15.2 (R Core Team, 2013) to generate 26 spectro-temporal features for each detection event. We used these features as predictors of whether or not a given signal was of interest. The features generated are listed in Table 2, along with explanations of their meanings. More detailed explanations of the features, as well as the source code used to calculate them, can be found in the documentation for the seewave R package. For features which require a spectrogram as their input, we used a 128-sample Hann FFT. When possible (Table 2), we generated spectrograms with overlapping frames, a procedure which can lead to smoother spectrograms.

Table 2
The spectro-temporal features generated for each event. The final 13 "specprop" measures are arrived at by converting the entire spectrum into a probability mass function. They are calculated only for the frequency band of the detector which created the event.

Feature name	Description	Spectrogram overlap
Event duration	The length of the event in seconds	n/a
Rugosity	A measurement of the RMS variation in amplitude from one sample to the next	n/a
Crest factor	The crest factor of the waveform	n/a
Temporal entropy	The entropy of the temporal envelope	n/a
Shannon entropy	The Shannon spectral entropy	0
Spectral flatness measure	An estimate of the flatness of a frequency spectrum—not band limited	0
Spectrum roughness	Roughness of a frequency spectrum	90%
Spectrum roughness—band limited	Spectrum roughness, computed only for frequency band of detector which created event.	90%
Autocorrelation mean	Mean fundamental frequency	0
Autocorrelation median	Median fundamental frequency	0
Autocorrelation standard error	Standard error of the fundamental frequency	0
Dominant frequency mean	Mean dominant frequency (i.e. highest-amplitude frequency)	90%
Dominant frequency standard error	Standard error of the mean dominant frequency	90%
Specprop mean	Mean frequency	0
Specprop standard deviation	Standard deviation of the frequency mean	0
Specprop SEM	Standard error of the frequency mean	0
Specprop median	Median frequency	0
Specprop mode	Mode frequency	0
Specprop Q25	First frequency quartile	0
Specprop Q75	Third frequency quartile	0
Specprop IQR	Frequency interquartile range	0
Specprop cent	Frequency centroid	0
Specprop skewness	Frequency skewness	0
Specprop kurtosis	Frequency kurtosis	0
Specprop SFM	Spectral flatness measure	0
Specprop SH	Shannon spectral entropy	0

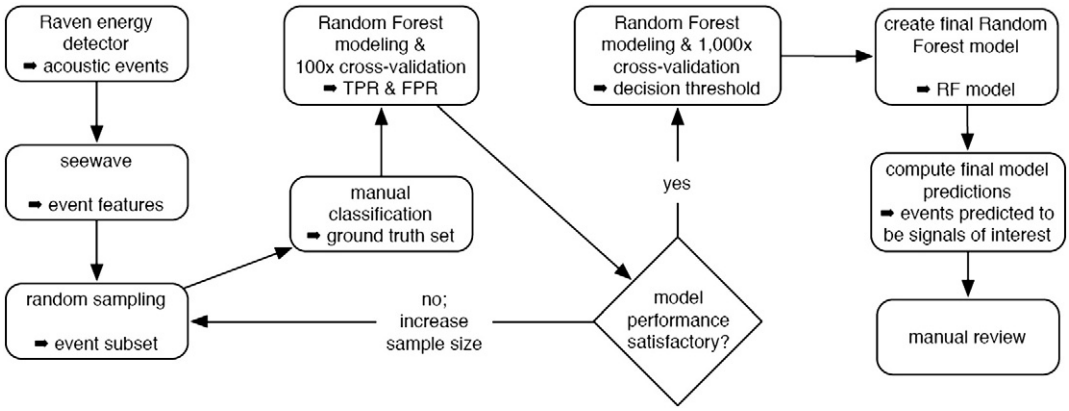

Fig. 1. Flowchart of entire analysis process. Arrows (➡) indicate the product of each step. As indicated in Table 3, this process typically results in reducing the number of events needing to be manually classified by 80% or more. Note that the final Random Forest model is created using all of the events from the ground truth set manually created earlier in the process.

Table 3
The numbers of events present in the phases of this workflow. In each case, more than 80% of the events were eliminated by the Random Forest model, and were never seen by an analyst.

Location	Detection band	Raw events detected	Events sampled (approx. % of total)	Events of interest in sample	Predicted events of interest outside of sample	Total events of interest	Percentage of events eliminated
Allegany	High	241,882	36,256 (15%)	28	10,381	194	80.8
Allegany	Low	93,517	14,044 (15%)	62	3733	727	81.0
Lewis	High	36,126	1740 (5%)	52	1842	1034	86.0
Lewis	Low	241,553	24,297 (10%)	72	11,205	872	85.4
Madison	High	75,849	7624 (10%)	126	3645	1099	85.2
Madison	Low	437,994	43,878 (10%)	162	19,732	2001	85.5
Ontario	High	48,369	4763 (10%)	57	2145	546	85.6
Ontario	Low	260,497	13,062 (5%)	89	11,358	1716	90.6
Tompkins	High	9436	480 (5%)	26	572	281	89.0
Tompkins	Low	109,751	15,457 (15%)	116	5094	1344	80.4
Wyoming	High	17,353	1752 (10%)	31	779	299	85.5
Wyoming	Low	138,506	6931 (5%)	125	7062	3245	89.9

2.4. Creating ground truth sets

All training, modeling, and evaluation were performed separately for each site-band combination using the events from a given site and band (Fig. 1). To create the ground truth data set for a model, we began by manually reviewing a small percentage of the relevant events, chosen randomly, and classifying them as signals of interest or false detections. We then built a model based on the truth set and evaluated its performance using cross-validation (Section 2.4.1). If performance of the model was insufficient, we iteratively increased the size of the ground truth data set by manually reviewing and classifying an additional 1–5% of the events, chosen randomly without replacement, for that location-band combination (Table 3). Each time the size of the truth set was increased, we computed a new model and evaluated its performance. Once cross-validation indicated that additional ground truth data would not improve a model (Section 2.4.1), we stopped adding to the truth set and proceeded to make a final model with which to classify the remaining events for that site and band (Section 2.5).

2.4.1. Evaluation of the ground truth set through cross-validation

We used hundred-fold cross-validation to evaluate whether the ground truth data set sufficiently captured the variety of calls and noise present at each recording site (Fig. 2). For each cross-validation run, we partitioned the nominal ground truth data set into a training set (70%) and test set (30%). For each training set, we used the R package randomForest version 4.6.7 (Liaw and Wiener, 2002) to create a Random Forest model for classifying events that represented signals of interest versus false detections. For a given event, each such model returned a score which represented the proportion of the trees in the Random Forest model which voted for that event as a signal of interest. To generate model predictions for a test event, one chooses a threshold value and those events scoring below the threshold are predicted to be a false detection. Those events scoring above the threshold are predicted to be events of interest.

We then evaluated the performance of the 100 Random Forest models by generating scores for each of the events in the test set for each model. It is possible to visualize the trade-off between precision

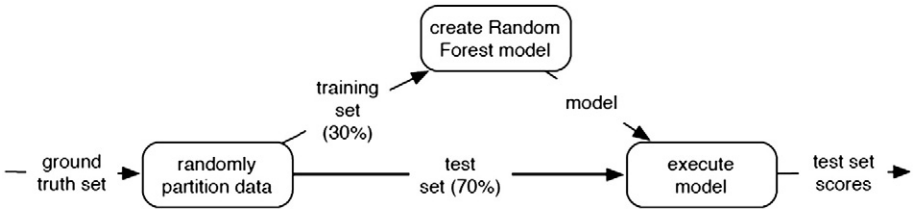

Fig. 2. Process for one cross-validation run. Each run begins with the same ground truth set but randomly partitions it, ultimately generating a unique model. The output of each run is a set of scores for the events in the test set.

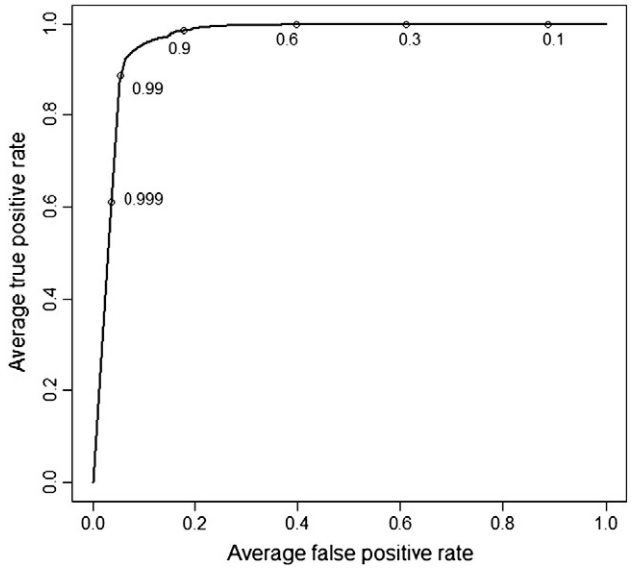

Fig. 3. ROC curve. The receiver operating characteristic curve generated by 100-fold cross-validation on an analyzed 10% of events detected in sound files recorded in Madison County, NY. The numbers printed along the curve represent different values of the decision threshold.

and recall at different thresholds by constructing a receiver operating characteristic (ROC) curve, as for example in Fig. 3. As a further example, ROC curves for the low-band detector at all six sites are shown in Fig. 4. We used the R package ROCR version 1.0–4 (Sing et al., 2012) to generate these curves. However, the ROC curve is not a precise way of determining the correct threshold to use in predictions. To numerically determine our threshold, we examined the distributions of the models' True Positive Rates (TPRs) at each threshold. We chose the minimum threshold which resulted in 95% or more of the Random Forest models achieving a TPR of 95% or greater. At this threshold, we expect a model to misclassify no more than 5% of the signals of interest.

We also monitored the False Positive Rate (FPR) for the cross-validation set by constructing a learning curve (e.g., Fig. 5). As we iteratively increased the size of the ground truth set, the FPR typically dropped rapidly at first, and then stabilized. Once the FPR had stabilized such that it was no longer decreasing significantly with the addition of new data, we considered our subsample of the data to be sufficient to characterize the variety of calls and noise present within the overall data set.

2.5. Final model generation and application

Once the FPR learning curve had stabilized and the introduction of new ground truth data no longer led to increased model precision, we generated a final model using the entire set of ground truth events as a training set. Using the same methods as above, but with more many more trials, we used thousand-fold cross-validation to determine a decision threshold for the final model expected to have at least a 95% probability of achieving a TPR of 95% or greater. We then used this final model to assign signal of interest versus false positive predictions to the remaining events for the given site and band. We discarded the events predicted to be false positives, and manually analyzed the remainder.

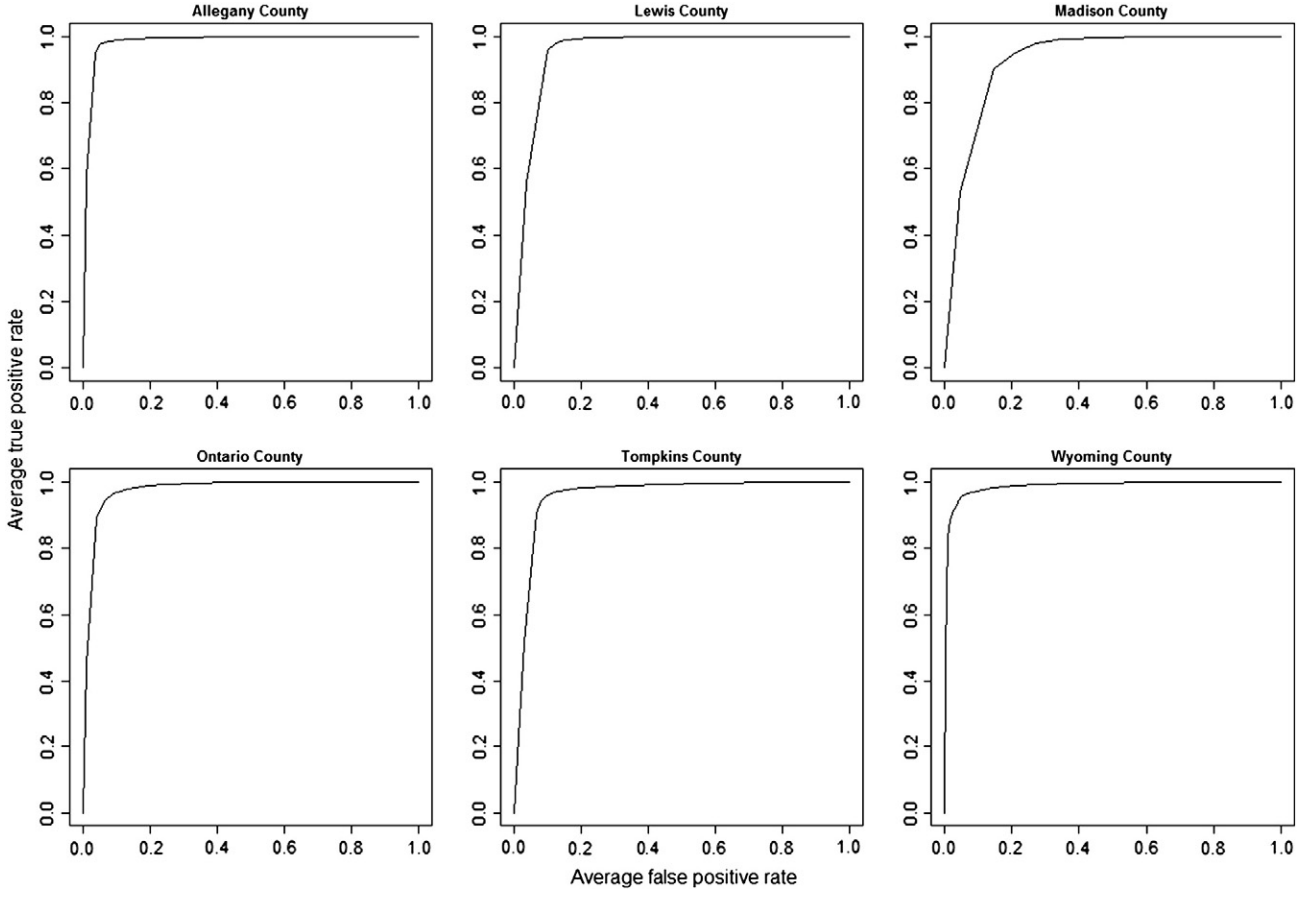

Fig. 4. ROC curves from all deployments. The receiver operating characteristic curves for the final models constructed for the results of the low-band detector at all six sites.

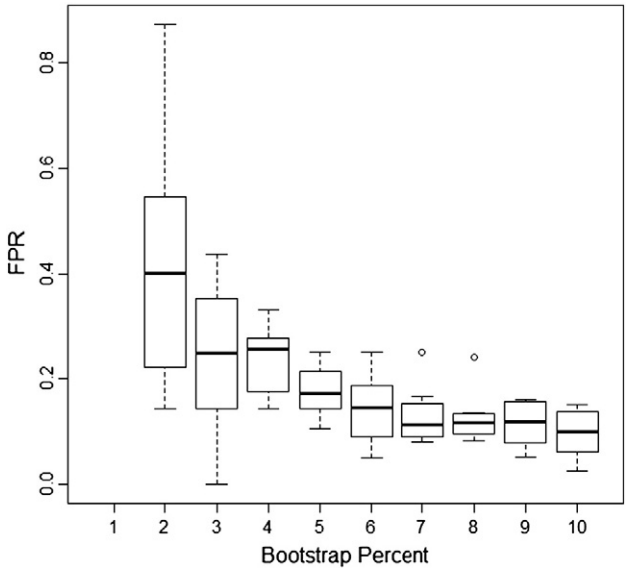

Fig. 5. Learning curve. The change in the mean false positive rate as subsampled training data was gradually added to the classifier training set for nocturnal recordings from Madison County, NY. The boxes and whiskers represent variance within the false positive rate of the population of 100 Random Forest models made during cross-validation. The x-axis indicates the proportion of events included in the training set.

Table 4
The mean decrease in Gini coefficient at nodes splitting on a given feature.

feature name	Mean decrease Gini
Event duration	3.71
Rugosity	9.38
Crest factor	8.56
Temporal entropy	17.47
Shannon entropy	6.42
Spectral flatness measure	6.44
Spectrum roughness	3.69
Spectrum roughness—band limited	4.32
Autocorrelation mean	3.33
Autocorrelation median	2.80
Autocorrelation standard error	2.43
Dominant frequency mean	5.73
Dominant frequency standard error	2.79
Specprop mean	5.72
Specprop standard deviation	5.45
Specprop SEM	4.02
Specprop median	5.28
Specprop mode	3.82
Specprop Q25	7.07
Specprop Q75	4.43
Specprop IQR	10.46
Specprop cent	6.02
Specprop skewness	7.09
Specprop kurtosis	5.06
Specprop SFM	2.97
Specprop SH	7.98

3. Results and discussion

Table 3 shows the numbers of events present in the phases of our analysis. In each case, the overall number of events which must be analyzed by humans was reduced by 80% or more.

The methods outlined here improve the efficiency of a common workflow in an NFC analysis. The goal in this workflow is to extract as many NFCs as possible from a sound stream, and this goal is never perfectly achieved, due to the cost and availability of analysts' labor. Since classifiers are intrinsically subject to a trade-off between precision and recall, any gain in precision (i.e., achieving a higher proportion of signals of interest in the data to be analyzed) is necessarily balanced by a reduction in recall. Within this limitation, we set and were able to achieve a goal of minimizing analyst labor while also ensuring that few legitimate detections would be discarded.

However, in order to make NFC analyses feasible, existing NFC detectors already make an implicit trade-off between precision and recall. Indeed, NFC detectors currently in use at the Cornell Lab of Ornithology, when tested against four night-long hand-browsed sound streams where all NFCs have been manually identified, have recall values in the range 16–61% for the low-band detector, and 21–54% for the high-band detector (A. Klingensmith, personal communication).

An important caution is that our goal of 95% or greater recall in our Random Forest classifier is for all types of NFCs combined, and NFCs are a diverse class of signals given by dozens of species. Just as band-limited energy detectors do not have the same detection probability for all species, it is possible that our classifiers may preferentially rank the NFCs of certain species higher than those of other species.

We make no claim of optimality for the acoustic feature set we used; in order to improve the efficiency of our analysis workflow, the features which seewave provides were sufficient. The features we used were of varying importance to the resulting model. The most important feature, as measured by the decrease in Gini coefficient at nodes splitting on that feature, was the temporal entropy (Table 4).

Many acoustic features not provided by seewave, such as Mel-frequency Cepstral Coefficients, can be calculated using other software, and such features have shown to have predictive power for bird syllables (e.g. Lee et al., 2008).

We make no claim of optimality for the Random Forest algorithm as a classifier; in order to improve the efficiency of our analysis workflow, the classification performance of Random Forest was sufficient. The simplicity of the algorithm, the availability of a polished open source package implementing it in R, and its victory in a 2011 semi-supervised learning competition (Sculley, 2011) were strong motivating factors in choosing it. However, other machine learning techniques, such as inductive semi-supervised learning (Yarowsky, 1995), might have more predictive power.

4. Conclusions

We have shown that an application of simple off-the-shelf machine learning techniques, using only free software, can lead to large gains in efficiency in the analysis of NFCs, with little deleterious effect upon the numbers of NFCs which can be extracted from a sound stream.

It is worth noting that the flightcallr software outlined here has also shown promise in other arenas of acoustic monitoring. For instance, a recent online classification challenge focused on distinguishing “up calls” of North Atlantic Right Whales from false positives (Kaggle, 2013). In that competition, the flightcallr software used here achieved an AUC (area under the ROC curve) score of 0.917. While this was well below the winning score of 0.983, it was achieved using the stock flightcallr package with no problem-specific tuning, and was a significant improvement from the existing benchmark of 0.722.

We developed this technique using the spectro-temporal features which are already included in the open source seewave R package, with minimal effort on our part to develop new features. There is likely potential for improving this technique through the addition of other meaningful acoustic features. We encourage researchers to contribute further open source software for generating meaningful acoustic features to the scientific community.

Acknowledgments

We thank A. Klingensmith, M. Powers, and R. Rohrbaugh for their support and patience during the development of this protocol. E. Griffiths, S. Keen, A. Klingensmith, and M. Powers provided helpful

comments on the manuscript. We gratefully acknowledge support from the Kenneth L. Harder Trust and the Scott and Karen Harder Family, as well as NYSERDA, for enabling this work.

References

Armitage, D.W., Ober, H.K., 2010. A comparison of supervised learning techniques in the classification of bat echolocation calls. Ecol. Inform. 5, 465–473.

Barkley, Y., Oswald, J.N., Carretta, J.C., Rankin, S., Rudd, A., Lammers, M.O., 2011. Comparison of real-time and post-cruise acoustic species identification of dolphin whistles using ROCCA (Real-time Odontocete Call Classification Algorithm). NOAA Technical Memorandum NMFS: NOAA-TM-NMFS-SWFSC-473.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Briggs, F., Fern, X.Z., Raich, R., 2009. Acoustic classification of bird species from syllables: an empirical study. Technical Report. (http://eecs.oregonstate.edu/research/bioacoustics).

Charif, R.A., Clark, C.W., Fristrup, K.M., 2004. Raven 1.2 User's Manual. Cornell Laboratory of Ornithology, Ithaca, NY.

eBird, 2012. eBird: An Online Database of Bird Distribution and Abundance [Web Application]. eBird, Ithaca, New York (Available: http://www.ebird.org. Accessed: October 31, 2012).

Evans, W.R., 2012. Avian acoustic monitoring study at the Maple Ridge Wind Project 2007–2008. Report No. 12–23 Prepared for the New York State Energy Research and Development Authority..

Evans, W.R., O'Brien, M., 2002. Flight Calls of Migratory Birds: Eastern North American Landbirds. [cd-rom] Oldbird, Ithaca, New York.

Farnsworth, A., 2005. Flight calls and their value for future ornithological studies and conservation projects. Auk 123, 733–746.

Farnsworth, A., Gauthreaux Jr., S.A., Van Blaricom, D., 2004. A comparison of nocturnal call counts of migrating birds and reflectivity measurements on Doppler radar (WSR-88D). J. Avian Biol. 35, 365–369.

Gagnon, F., Belisle, M., Ibarzabal, J., Vaillancourt, P., Savard, J., 2010. A comparison between nocturnal aural counts of passerines and radar reflectivity from a Canadian weather surveillance radar. Auk 127, 119–128.

Henderson, E.E., Hildebrand, J.A., Smith, M.H., 2011. Classification of behavior using vocalizations of Pacific white-sided dolphins (*Lagenorhynchus obliquidens*). J. Acoust. Soc. Am. 130, 557–567.

Kaggle, 2013. The Marinexplore and Cornell University Whale Detection Challenge. http://www.kaggle.com/c/whale-detection-challenge/ (Accessed 2013-06-21).

Kunz, T.H., Arnett, E.B., Cooper, B.M., Erickson, W.P., Larkin, R.P., Mabee, T., Morrison, M.L., Strickland, M.D., Szewczak, J.M., 2007. Assessing impacts of wind-energy development on nocturnally active birds and bats: a guidance document. J. Wildl. Manag. 71, 2449–2486.

Lee, C.-H., Han, C.-C., Chuang, C.-C., 2008. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. IEEE Trans. Audio Speech Lang. Process. 16, 1541–1550.

Liaw, A., Wiener, M., 2002. Classification and regression by randomForest. R News 2, 18–22.

Powers, M., Klingensmith, A., Farnsworth, A., Ross, J., MacGillivray, M., Rohrbaugh, R., 2013. Acoustic monitoring of migrant landbirds for: balancing wind and wildlife: new data and tools to improve wind project siting for biodiversity conservation. Report Prepared for the New York State Energy Research and Development Authority.

R Core Team, 2013. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (URL http://www.R-project.org.).

Ross, J., 2013. flightcallr: classify night flight calls based on acoustic measurements. R Package Version 0.04. (URL http://r-forge.r-project.org/projects/flightcallr/).

Sculley, D., 2011. Results from a semi-supervised feature learning competition. Presented in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Dec 2011 (URL http://www.eecs.tufts.edu/~dsculley/papers/semisupervised-feature-learning-competition.pdf).

Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T., 2012. ROCR: visualizing the performance of scoring classifiers. R Package Version 1.0-4. (URL http://CRAN.R-project.org/package=ROCR).

Sueur, J., Aubin, T., Simonis, C., 2008. Seewave: a free modular tool for sound analysis and synthesis. Bioacoustics 18, 213–226.

Yarowsky, D., 1995. Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, MA, pp. 189–196.