Technical Report (Not Peer Reviewed): Acoustic Classification of Bird Species from Syllables: an Empirical Study

Forrest Briggs
Oregon State University
briggsf@onid.orst.edu

Xiaoli Fern
Oregon State University
xfern@eecs.oregonstate.edu

Raviv Raich
Oregon State University
raich@eecs.oregonstate.edu

Abstract

In order to automatically extract ecologically useful information from audio recordings of birds, we need fast and accurate algorithms to classify bird sounds. We conduct a large-scale empirical study to evaluate algorithms for classifying bird species from audio using combinations of 3 feature sets (Mel-frequency cepstral coefficients, average spectra, and noise-robust measurements), with 10 classifiers including support vector machines and AdaBoost (with J48), on a 2.49 Gb data set consisting of recordings of 20 species of bird from the Cornell Macaulay library. Using implementations from the Weka machine learning system, Random Forest is always close to AdaBoost and usually more accurate than SVM, while being an order of magnitude faster than both.

1 Introduction

Our goal is to develop algorithms that classify bird sounds according to species. We intend to use such a classifier to automatically extract information about patterns of bird activity from a large data set of audio collected in the field.

Spectrograms are widely used for understanding bird sounds [4]. Figure 1 shows the spectrogram for a one second recording of a Mountain Chickadee. A spectrogram is a graph of the intensity of a sound as a function of time and frequency. Bird sound often has a gramatical structure, in which the basic building blocks are called syllables [4]. In this study, we consider algorithms for classifying syllables by species. The boxes in Fig. 1 indicate syllables. A syllable is basically a single distinct utterance by a bird.

We evaluate three types of features to describe syllables: Mel-frequency cepstral coefficients (MFCCs), which among the most widely used features in audio classification, average spectra (i.e. average rows of Fig. 1), and noise-robust measurements, which have been used for classifying sounds made by marine mammals [22], but not extensively evaluated on bird sound. We compare the accuracy of 10 standard algorithms for supervised learning on these features (see Sec. 4). We are not aware of any prior work on bird species classification that employs Adaboost, Bagging or Random Forest classifiers, although SVMs [10], neural nets [21], and nearest-neighbor classifiers [20, 14, 25, 6] are used.

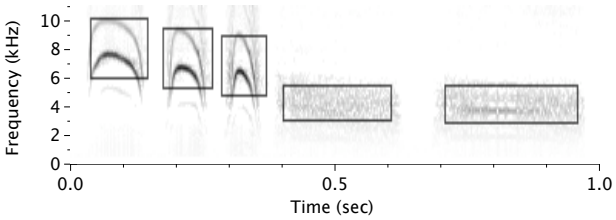

Figure 1. The spectrogram for one second of a recording from a Mountain Chickadee. The boxes indicate syllables detected by the algorithm given in Sec. 3.3.

The experiments presented herein involve 20 species of bird and over 30,000 syllables, which makes this a much larger scale study than most prior work. We conclude that Random Forest offers clear advantages over SVM and AdaBoost with J48 on a problem of this size, because it provides the best tradeoff between runtime and accuracy. Furthermore, the average spectrum representation of syllables gives better classification accuracy than MFCCs, which are more widely used (although average spectrum is a higher-dimensional feature, and thus, slower).

2 Bird Species Classification: Background

In this section, we provide background on audio classification, and discuss prior work on bird species recognition.

Signals, Frames and Spectrograms Audio data consists of a time-series of samples, denoted $s(1), s(2), \ldots s(n)$.

Sound is sampled at a particular rate f_s, which specifies how many samples are in one second. It is easier to identify patterns in audio, particularly bird sound, after transforming a series of samples into a spectrogram [4]. To do so, a signal is first divided into overlapping frames, each of which consists of a sequence of consecutive samples. The fast Fourier Transform (FFT) is then applied to each frame to obtain their spectra, which describe the intensity of the sound within each frame, as a function of frequency. The spectrum of a frame consists of a list of complex Fourier coefficients returned by the FFT. We use the magnitudes of these complex numbers, which are referred to as a the magnitude spectrum. The FFT produces a spectrum for each frame in a signal; a spectrogram is a graph of the spectra of a series of frames. Figure 1 shows an example spectrogram.

2.1 Features

This section explains the calculation of three types of features to describe syllables: average spectra, average MFCCs, and noise-robust measurements. We defer discussion of how syllables are detected to Sec. 3.3.

Average Spectra To calculate average spectra and MFCCs, consider the start and end time (in frames) of a syllable, and the spectrogram within that range (note that we do not use the frequency bounds depicted in Fig. 1 when computing these features). If the start time is t_{min} and the end time is t_{max}, the average spectrum is $Avg(j) = \frac{1}{t_{max}-t_{min}} \sum_{i=t_{min}}^{t_{max}} Q(i,j)$, where $Q(i,j)$ is a spectrogram after normalization and noise-reduction (see Sec. 3.2).

Mel-Frequency Cepstral Coefficients Mel-frequency cepstral coefficients [8] (MFCCs) are one of the most widely used features in audio classification. The general idea is to first compute Mel-frequency coefficients (MFCs), which are like the magnitude spectrum, but in units of mels rather than Hz (mels correspond more closely with human perception of pitch [26]). MFCs are computed by applying a collection of triangle filters to the magnitude spectrum; the MFCs are the response of each filter. The filters are evenly spaced in the mel scale. MFCCs are the result of applying the discrete cosine transform (DCT) to the log of the MFCs. Like spectra, MFCCs describe a single frame, and are averaged over the duration of a syllable.

Noise-Robust Measurements Mellinger et al. [22] proposed a set of noise-robust features for classification of vocalizations made by marine mammals. Unlike average spectra and MFCCs which require just the start and end time of a syllable, noise-robust measurements also involve a minimum and maximum frequency, which define a box in spectrogram around a syllable. Using the notation in [22], features describing a syllable are denoted as $M1, M2, M3, \ldots M20$ and $Q1, Q3$. Table 1 lists the combinations of these features that we use (18 in total). To normalize the ranges of these features, we divide a feature with units of time by the syllable's duration to make it unitless, and similarly, we divide features with units of frequency by the syllable bandwidth.

$M3$	Min. Frequency (kHz)
$M4$	Max. Frequency (kHz)
$M5$	Duration (sec)
$M6$	Bandwidth (kHz)
$M7/M5$	Median Time / Duration
$M11/M6$	Median Frequency / Bandwidth
$Q1_{time}/M5$	Temporal Low Quartile / Duration
$Q3_{time}/M5$	Temporal High Quartile / Duration
$Q1_{freq}/M6$	Frequency Low Quartile / Bandwidth
$Q3_{freq}/M6$	Frequency High Quartile / Bandwidth
$M8/M5$	Temporal Interquartile Range / Duration
$M12/M6$	Spectral Interquartile Range / Bandwidth
$M10/M5$	Temporal Asymmetry / Duration
$M14/M6$	Spectral Asymmetry / Duration
$M16$	Relative Time of Peak Cell Intensity
$M18$	Relative Time of Peak Overall Intensity
$M19/M6$	Frequency of Peak Cell Intensity / Bandwidth
$M20/M6$	Frequency of Overall Peak Intensity / Bandwidth

Table 1. The features we use from Melinger's noise-robust measurements.

2.2 Related Work

Many authors have proposed systems for classifying bird species from audio recordings [21, 25, 10, 20, 15, 6, 14], which generally work by first segmenting audio into syllables, computing features such as Mel-frequency cepstral coefficients [8] (MFCCs) for those syllables, then applying a standard classifier such as neural nets [21] or support vector machines [10].

Most algorithms for detecting syllables work by computing an energy envelope for a sound, comparing it to a threshold, then predicting syllables where the energy is above the threshold [15, 25, 10, 24, 24].

After running a segmentation algorithm to identify syllables, systems for bird species recognition extract acoustic features to characterize the syllables in a way that can be used with machine learning algorithms for classification. Linear Predictive Coding (LPC) [20, 18, 6, 21] and Mel-frequency cepstral coefficients (MFCCs) [8, 13, 11, 26, 20, 18, 19, 25, 10, 6], are common in analysis of speech and music, and are amongst the most widely used features to

describe syllables of bird sound. Features such as LPCs and MFCCs describe individual frames of sound; to characterize a syllable as a whole, a standard approach is to average the frame-level features [10, 20, 25] . Other features that have been used to characterize syllables include spectral peak tracking and analysis-by-synthesis / overlap-add (ABS/OLA) [14, 15, 25, 6], Wavelets [24], and 'descriptive parameters' such as bandwidth (one of the noise-robust measurements), zero-crossing rate and spectral flux [25, 10].

Some prior work is concerned with classification of bird species from individual syllables [10], while other work is also concerned with identifying species from songs composed of sequences of syllables [25, 18]. The algorithms that have been applied to classifying syllables include nearest-neighbor and distance based classifiers [20, 14, 25, 6], multi-layer perceptrons [24, 21], self-organizing maps [24], and support vector machines [10].

3 Experimental Setup

This section explains the experimental setup we used to evaluate the features and classifiers introduced in Sec. 2.1 and 3.5. The goal is to measure the accuracy of these classifiers in predicting bird species from a single syllable, described by a fixed-length feature vector.

3.1 Data

We have 2.49 Gb of audio recordings of 20 species of bird (see Fig. 2), from the Cornell Macualay library of ornithology, encoded as mono 8-bit WAV files, with sampling rate $f_s = 44.1kHz$. There are 7 files for each species, and each file is between one minute and 20 minutes long. These recordings were collected over several decades, mostly in the western United States. Each recording is labeled with a species of bird. Most of the sounds in a recording are from one or more birds of this species, although there are sometimes other noises including different species of bird, car sounds, and human voices. We manually removed most portions of recordings containing human voice.

3.2 Preprocessing

In Sec. 2, we gave a general description of audio signals and features to describe them. In this section, we elaborate on more details and parameters specific to our implementation.

Chunks and Spectrograms We divide the original audio files into 10 second 'chunks', resulting in 2,963 chunks, for a total of 493.8 minutes of audio. Working with small chunks of sound reduces the amount of memory needed for computations. To compute a spectrogram for each chunk,

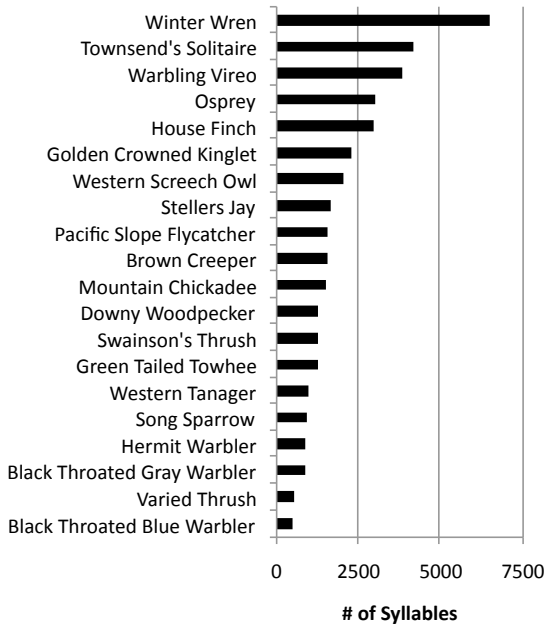

Figure 2. The number of syllables for each species in the data set.

we divide it into frames of 256 samples, with 50% overlap. Running the FFT on a frame of 256 samples gives a spectrum with 128 Fourier coefficients, but to decrease computation time, we use only the lower 64 coefficients. To reduce noise, we set all values for the lowest 3 coefficients to zero. These two operations are equivelent to filtering out sound below $517.97Hz$ and above $11.05kHz$. Most bird sound is in this range.

Normalization & Noise Reduction Although noise and intensity levels vary considerably between recordings, we can assume they are relatively constant over the duration of a 10 second chunk. Within each chunk, we normalize the spectrogram by dividing all cells by the value of the maximum cell, and remove the stationary noise by subtracting the average spectrum (similar to spectral subtraction [1]). If the magnitude spectrum for frame i and frequency bin j is $M(i,j)$, we compute the normalized spectrogram as $N(i,j) = M(i,j)/\max_{i,j} M(i,j)$. Next, we compute the average spectrum over the duration of a chunk, $A(j) = \frac{1}{n}\sum_{i=1}^{n} N(i,j)$, where n is the number of frames in the chunk, then subtract this vector from the normalized spectrum for each frame, to get the noise-reduced spectrogram, $R(i,j) = \max(0, N(i,j) - A(j))$. Here j ranges from 1 to the number of elements in the spectrum. We use the values of R directly in the calculation of MFCCs (see Sec. 2.1), but for detecting syllables and computing average spectra and noise-robust measurements, we use $Q(i,j) = \sqrt{R(i,j)}$ in-

stead; this is an ad-hock adjustment to boost contrast between background noise and signal.[1] The spectrogram in Fig. 1 is a direct representation of $Q(i, j)$, with white corresponding 0 and black corresponding 1.

3.3 Detecting Syllables

Similar to Fagerlund [10] and Härmä et al. [25], we use an energy based segmentation algorithm to identify the start and end times of syllables. We compute the energy envelope from the de-noised spectrogram as $E(i) = \sum_{j=1}^{64} Q(i, j)$. After computing the energy envelope, we smooth it by applying a uniform convolution with a neighborhood size of 7 frames (i.e. each $E(i)$ is replaced by the average of 15 elements of E, which includes 7 elements on either side of i, and element i). Then we normalize the energy by dividing each element by the value of the maximum. Next, we compute the average energy $E_{avg} = \frac{1}{n} \sum_{i=1}^{n} E(i)$, and a threshold $T = \frac{3}{2} E_{avg}$. Syllables are defined by the regions where $E(i) \geq T$, with their start and end frames being the points where $E(i)$ crosses T. The constant $\frac{3}{2}$ was chosen by visual inspection of segmentation results with various thresholds.

The noise-robust measurements $M1$ and $M2$ correspond to the start and end time of a syllable. The algorithm described by Mellinger [22] requires that a human first provide a rough annotation box around a syllable to compute $M1$ and $M2$. We instead use the start and end-frames identified by the energy-threshold algorithm above. Noise-robust measurements also require a minimum and maximum frequency, which combined with the start and end time for a syllable, form a box (see Fig. 1). We compute frequency bounds for syllables ($M3$ and $M4$) following the algorithms given in [22], starting from $Q(i, j)$, i.e. the spectrogram after normalization, noise-reduction and contrast enhancement.[2]

3.4 Implementation of MFCCs

Our implementation of MFCCs is based on the description provided by Ganchev et al. [13] of the MFCCs computed in the Cambridge Hidden Markov Models Toolkit (for MATLAB), known as HTK [28]. In our implementation, the first triangle filter is centered at $1033.59Hz$, and the last filter's highest value is $22.1kHz$.[3] We use 24 filters, resulting in 24 MFCs, then take only the first 12 elements of the output of the DCT as a feature to describe each frame (similar to Fagerlund [10]). Then, we average the MFCCs for each frame within a syllable to get a single set of MFCCs that describe the syllable as a whole, which results in a 12-dimensional feature vector. Note that for computing MFCCs, we use the de-noised spectrogram $R(i, j)$, without the contrast boost.

3.5 Classifiers

Using the features described in Sec. 2.1, we evaluate a variety of algorithms for supervised learning:

- Naive Bayes
- Logistic Regression
- Multi-Layer Perceptron (MLP) [24, 21]
- k-Nearest-Neighbors [20, 14, 25, 6]
- J48 (a pruned decision tree based on C4.5 [23])
- Adaboost [12] applied to J48
- Bagging [2] applied to J48.
- Random Forest [3]
- Support vector machines [7, 10]

The classifiers we use in this experiment are part of Weka [27], an open-source software package written in Java that provides a collection of widely used algorithms for machine learning. Table 2 lists the classifiers that we used, and the command to Weka specifying the parameters for each classifier, as well as the results, which are discussed in Sec. 4.

Because it is much faster than the basic implementation of SVMs in Weka, we used WLSVM [9], which integrates LIBSVM [5] into Weka. Following Fagerlund [10], and the recommendations of Hsu, Chang and Lin [16], we use a radial basis function kernel, and optimize the SVM parameter C and the kernel parameter γ, by grid search. We evaluate the SVM at all combinations of C and γ in $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. To handle multiple classes (in our case, species), we use the one-against-one voting scheme [17].[4]

[1] Visually, we find that spectrograms show clearer distinction between bird sound and noise with this modification.

[2] For calculating $M3$ and $M4$, Mellinger used an energy threshold of 90% instead, but we visually we preferred the results with 75%.

[3] Following an exact implementation of the filters described by Ganchev et al. [13], we got aliased triangle filters because some were narrower than a single spectrum bin, which caused artifacts in the MFCs. To fix this problem, we numerically integrate the triangle filter function over the range of each bin. Many other implementations of MFCCs work with lower sampling frequencies [13], so we suspect this problem is related to working with sound sampled at $44.1kHz$, as well as our choice of values for f_{low} and f_{high}.

[4] Fagerlund constructs a decision tree of binary SVMs to handle multiple classes; one-against-one voting is recommended over this approach by the authors of LIBSVM.

Classifier	Spectrum	MFCCs	Noise-Robust	Weka command
Naive Bayes	27.09%	31.08%	26.43%	weka.classifiers.bayes.NaiveBayes
Logistic Regression	27.63%	30.46%	29.76%	weka.classifiers.functions.Logistic -R 1.0E-8 -M 100
MLP	33.80%	35.16%	34.13%	weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 100 -V 0 -S 0 -E 20 -H a
1-Nearest-Neighbor	38.72%	35.88%	30.98%	weka.classifiers.lazy.IB1
5-Nearest-Neighbor	39.24%	36.59%	31.87%	weka.classifiers.lazy.IBk -K 5 -W 0 -A
J48	32.26%	29.76%	30.73%	weka.classifiers.trees.J48 -C 0.25 -M 2
Adaboost with J48	50.29%	39.18%	40.34%	weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 100 -W weka.classifiers.trees.J48 – -C 0.25 -M 2
Bagging with J48	46.35%	37.15%	39.39%	weka.classifiers.meta.Bagging -P 100 -S 1 -I 100 -W weka.classifiers.trees.J48 – -C 0.25 -M 2
Random Forest	48.58%	39.74%	40.34%	weka.classifiers.trees.RandomForest -I 100 -K 0 -S 1
SVM	42.30%	41.49%	38.61%	weka.classifiers.functions.LibSVM -S 0 -K 2 -D 0 -G γ -R 0.0 -N 0.0 -M 512.0 -C C -E 0.0010 -P 0.1 -Z

Table 2. The accuracy for each classifier and feature set, and the Weka command for the classifier.

3.6 Cross Validation

There are 7 representative recordings from each species, i.e. wren1.wav, ..., wren7.wav, owl1.wav, ..., owl7.wav, etc. These files generally contain recordings of one or more individual birds repeatedly producing a similar song or call. If we train and test a classifier on examples that come from the same recording, classification is relatively easy, because calls from the same individual bird over a short period of time are often very similar. We use a setup that ensures that classifiers must be general across individuals of the same species, rather than just learning to classify specific birds in the training set. Somervuo, Härmä and Fagerlund use similar 'individual independent' cross validation [25]. We run leave-one-out cross validation at the scale of files; there are 7 folds which correspond to 7 audio files for each species. In each fold, all syllables in one file from each species are set aide for testing, and syllables in the remaining six files are used for training.

3.7 Unbalanced Training Examples

Some species, such as the Winter Wren, and Townsend's Solitaire, produce many more syllables per unit time than other species (see Fig. 2). Consequently, the training and test data sets contain more syllables for these species than the others. Classifiers trained with all available data are strongly biased towards these two majority classes. To avoid this bias, we do not use all available training data, but instead use a balanced subset of training examples. In each fold of cross-validation, we randomly sample 250 representative training examples from each species (with replacement), for a total of 5,000 training examples for 20 species.

3.8 Measuring Accuracy

Although we only use 5,000 training syllables in each fold of cross-validation, we still classify all 39,747 test syllables over the 7 folds. If we computed accuracy for a classifier as the fraction of syllables it correctly predicts, this measure will mostly characterize accuracy for the Winter Wren and Townsend's Solitaire. Instead, we calculate the classifier's accuracy for each species, then take the average of the per-species accuracies to get a single overall average accuracy.

4 Results

Table 2 lists the average per-class accuracy for each combination of features and classifier. AdaBoost with average spectra features was the most accurate overall, but Random Forest was almost as accurate. In Weka's implementation, Random Forest was about 10 times faster than AdaBoost with J48,[5] which we think is because the trees constructed by J48 are pruned with a relatively complex algorithm, whereas the trees in Random Forest are not pruned, and deal with subsets of features during induction. Random Forest in Weka was 20 times faster than WLSVM, which is primarily due to the need to optimize the SVM parameters C and γ, and because one-against-one voting requires $\binom{20}{2} = 190$ binary SVMs. These results suggest that Random Forest provides the best tradeoff between accuracy and runtime.

Classifiers using average spectra features were generally the most accurate, but this is the highest-dimensional feature (average spectra are 64 dimensional, MFCCs are 12 dimensional, and noise-robust measurements are 18 dimensional).

Somervuo, Härmä and Fagerlund [25] conducted an evaluation of bird species classification from syllables with 14 species, using MFCCs and other features, and nearest-neighbors classifiers with and without dynamic time warping (a technique for comparing variable length sequences

[5] Tests were run on a MacBook Pro with dual 2.53 GHz Intel Core 2 Duo processors and 4GB of 1067 MHz DDR3 ram. Runtime includes both training and testing.

of features), and achieved accuracies around 50%. They used an energy-threshold segmentation algorithm and an individual-indepented cross-validation setup, which makes our experiments roughly comparable. However, we observe that the number of species in a classification problem greatly affects its difficulty, so achieving about 50% accuracy on a problem with 20 species suggest that our approach improves on prior work (although other factors confound the comparison, such as the number of training examples and which species are included). Fagerlund's more recent work on syllable classification with SVMs cannot be directly compared, because there are only 5 species being classified [10].

5 Conclusion

The experimental results of this study suggest that Random Forest offers the best tradeoff between accuracy and runtime amongst a variety of other classifiers including SVMs. For future work in bird species classification, we suggest the use of average spectrum features to represent syllables, and Random Forest for classification.

References

[1] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

[2] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] C. Catchpole, P. Slater, and N. Mann. *Bird song: biological themes and variations*. Cambridge Univ Pr, 2003.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[6] Z. Chen and R. C. Maher. Semi-automatic classification of bird vocalizations using spectral peak tracks. *J Acoust Soc Am*, 120(5 Pt 1):2974–2984, November 2006.

[7] C. Cortes and V. Vapnik. Support vector networks. In *Machine Learning*, pages 273–297, 1995.

[8] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Readings in speech recognition*, pages 65–74, 1990.

[9] Y. EL-Manzalawy and V. Honavar. *WLSVM: Integrating LibSVM into Weka Environment*, 2005.

[10] S. Fagerlund. Bird species recognition using support vector machines. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.

[11] Z. Fang, Z. Guoliang, and S. Zhanjiang. Comparison of different implementations of mfcc. *J. Comput. Sci. Technol.*, 16(6):582–589, November 2001.

[12] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.

[13] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various mfcc implementations on the speaker verification task. In *in Proc. of the SPECOM-2005*, pages 191–194, 2005.

[14] A. Härmä. Automatic identification of bird species based on sinusoidal modeling of syllables. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 5, pages V–545–8 vol.5, 2003.

[15] A. Härmä and P. Somervuo. Classification of the harmonic structure in bird vocalization. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 5, pages V–701–4 vol.5, 2004.

[16] C. Hsu, C. Chang, C. Lin, et al. A practical guide to support vector classification, 2003.

[17] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[18] J. A. Kogan and D. Margoliash. Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study. *The Journal of the Acoustical Society of America*, 103(4):2185–2196, 1998.

[19] C. Kwan, G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, and K. Ho. Bird classification algorithms: Theory and experimental results. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 289–292, 2004.

[20] C.-H. Lee, Y.-K. Lee, and R.-Z. Huang. Automatic recognition of bird songs using cepstral coefficients. *Journal of Information Technology and Applications*, 1(1):17 – 23, 2006.

[21] A. McIlraith and H. Card. Birdsong recognition using backpropagation and multivariate statistics. *IEEE Transactions on Signal Processing*, 45(11):2740–2748, 1997.

[22] D. Mellinger and J. W. Bradbury. Acoustic measurement of marine mammal sounds in noisy environments. In *Proc. International Conference on Underwater Acoustic Measurements: Technologies and Results*, pages 273–280, 2007.

[23] J. Quinlan. *C4. 5: programs for machine learning*. Morgan Kaufmann, 1993.

[24] A. Selin, J. Turunen, and J. Tanttu. Wavelets in recognition of bird sounds. *EURASIP Journal on Advances in Signal Processing*, 2007:1–9, 2007.

[25] P. Somervuo, A. Härmä, and S. Fagerlund. Parametric representations of bird sounds for automatic species recognition. In *IEEE Transactinos on Audio, Speed, and Language Processing*, volume 14. IEEE Press, 2006.

[26] J. Volkmann, S. S. Stevens, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):208–208, 1937.

[27] I. Witten and E. Frank. Data mining: practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Record*, 31(1):76–77, 2002.

[28] S. Young. The hidden markov model toolkit. *Entropic Cambridge Research Laboratory, Ltd*, 2:2–44, 1995.