The Orchive : A system for the annotation and semi-automatic analysis of a large collection of bioacoustic recordings.

by

Steven Ness
B.Sc., University of Alberta, 1994
M.Sc., University of Victoria, 2010

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in the Department of Computer Science

© Steven R. Ness, 2013
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying
or other means, without the permission of the author.

The Orchive : A system for the annotation and semi-automatic analysis of a large collection of bioacoustic recordings.

by

Steven Ness
B.Sc., University of Alberta, 1994
M.Sc., University of Victoria, 2010

Supervisory Committee

Dr. G. Tzanetakis, Supervisor
(Department of Computer Science)

Dr. P. Storey, Departmental Member
(Department of Computer Science)

Dr. P. Driessen, Departmental Member
(Department of Electrical Engineering)

Supervisory Committee

Dr. G. Tzanetakis, Supervisor
(Department of Computer Science)

Dr. P. Storey, Departmental Member
(Department of Computer Science)

Dr. P. Driessen, Departmental Member
(Department of Electrical Engineering)

ABSTRACT

Advances in computer technology have allowed for the collection, digitization and automated processing of huge archives of bioacoustic sound. Many of the tools previously used in bioacoustics work well with small to medium sized audio collections, but are challenged when large collections of tens of terabytes to petabyte size. In this thesis, a system to allow researchers to listen, view and annotate these collections. This system also provides functionality to allow for the semi-automatic labelling of these recordings using a combination of advanced audio feature extraction and machine learning tools. In addition, this system allows citizen scientists to participate in the process of annotating these large archives using a casual game metaphor. In this thesis, the use of this system to annotate a large audio archive called the Orchive will be evaluated. The Orchive contains over 20,000 hours of orca vocalizations collected over the course of 30 years, and represents one of the largest continuous collections of bioacoustic recordings in the world. The effectiveness of this for the task of deriving knowledge from these recordings will be evaluated and results showing the utility of this system will be shown.

Contents

Supervisory Committee **ii**

Abstract **iii**

Table of Contents **iv**

List of Tables **vi**

List of Figures **xi**

Acknowledgments **xiv**

Dedication **xv**

1 Introduction **1**

1.1 Contributions 3

1.2 Large Bioacoustic Archives 4

1.3 MIR and Bioacoustics 7

1.3.1 Relevance of this work 9

1.4 Scope of this work 12

2 Related Work **14**

2.1 Bioacoustics 15

2.1.1 Orcas 18

2.1.2 Orcas - Algorithms 22

2.2 Systems for working with large audio corpora 28

2.3 Intelligence Augmentation 29

2.4 Citizen Science 34

3 Software and Systems **37**

3.1 Choice of Platform 38
3.2 Orchive Version 1.0 39
3.3 Orchive Version 2.0 43
3.4 System Overview 45
3.5 Machine Learning 49
3.6 Distributed Computing 50

4 Case Study **51**
4.1 Introduction 51
4.2 The Orchive 51

5 Evaluation **56**
5.1 Orca/Background/Voice Segmentation 57
5.2 Call classification 84
5.3 Symbolic Approximation of Pitch Contours 98
5.4 Citizen Science Interface 107
5.5 Expert Interface 114

6 Conclusions and Future Work **126**
6.1 Conclusions 126
6.2 Future work 128

7 Web Links **130**

8 Publications from this Research **131**

9 Glossary **133**

Bibliography **136**

10 Appendix A **154**
10.1 Alberta Biodiversity Monitoring Institute 154
10.1.1 Birds and Biodiversity 155
10.1.2 ABMI 158
10.1.3 Bird song identification 163

List of Tables

Table 5.1 Table of MFCC results with different window sizes using the LIBLINEAR classifier. In this and subsequent tables, "ws" refers to the size of the FFT window in samples, "hp" refers the hop size between sequent FFT frames in samples, and "mem" refers to the size of the texture window in frames. 59
Table 5.2 Table of MFCC results with different numbers of MFCC coefficients using the LIBLINEAR classifier. 61
Table 5.3 Table of LIBLINEAR parameters 63
Table 5.4 A table showing the results of doing a parameter search over a wide number of values of C and using the different solvers in the LIBLINEAR package. Please note that approximately 6x as many results as would fit here were also calculated, but were omitted due to space restrictions. These omitted results showed approximately the same behaviour as those above. 63
Table 5.5 Table showing the classification accuracy and timing when using different FFT window sizes on MFCC features using a linear SVM kernel. 68
Table 5.6 Table showing the effect of window size on classification performance with LIBLINEAR using the Root Mean Square (RMS) energy of a signal as a feature. 68
Table 5.7 Table showing the effect of using combinations of different statistical measures of the spectrum of a signal on classification performance with LIBLINEAR. 69

Table 5.8 Table showing the classification performance of LIBLINEAR with different combinations of Chroma, Spectral Crest Factor (SCF) and Spectral Flatness Measure (SFM). Locations in the table with NC represent "Not Completed" combinations where the size of the resulting feature file was greater than the maximum 3GB file size allowed by LIBLINEAR. 70
Table 5.9 Table showing the result of using the YIN pitch estimator as a feature for input to the LIBLINEAR SVM classifier with different window sizes and hop sizes. 70
Table 5.10 Table showing the use of all audio features described above as input to the LIBLINEAR package. In this table "css" refers to the combination of Chroma, SCF and SFM features. 71
Table 5.11 Table showing the results of different amounts of downsampling on classification performance with the LIBLINEAR SVM using MFCC audio features. Please note that approximately 3x as much data points as this were calculated but were omitted due to space. These omitted data points showed the same approximate performance as those shown here. 72
Table 5.12 Table showing the effects of changing different parameters of the polynomial kernel on the classification performance of LibSVM. The features used in this table include the collection of all audio features described above. In the first half of the table, different degrees of polynomials are tested, and in the second half different values of C and G are tested. 74
Table 5.13 Results showing the effect of changing the values of coef0 and gamma with LibSVM when using a sigmoid kernel. Please note that approximately 3x as many results were calculated but were omitted as they had approximately the same performance as those shown above. 75
Table 5.14 Table showing the results of using the Radial Basis Function kernel under LibSVM using different values of the Cost and Gamma parameters. Results calculated above use all the audio features described earlier. 77

Table 5.15 Table showing results of Logistic Regression classifier in the Weka software package with different values of the ridge parameters for the ridge in the log-likelihood function. 78
Table 5.16 Table showing results of Naive Bayes classifier in the Weka package with the -D parameter which corresponds to the use of supervised discretization to process numeric attribute. The results in this table were calculated with a combination of all the previously described audio features. 79
Table 5.17 Table showing results of J48 decision tree classifier with different values of all adjustable parameters and a combination of all audio features. In this table, the parameter "-A" enables Laplace smoothing for predicted probabilities, "-C" sets the onfidence threshold for pruning, "-L" turns on functionality to not clean up after the tree has been built, "-M" sets the minimum number of instances per leaf, "-N" specifies the number of folds for reduced error pruning, "-R" turns on reduced error pruning, "-S" disables subtree raising, and "-U" uses an unpruned tree. . . . 81
Table 5.18 Table showing results of random forest classifier with different values of the number of trees to build (-I) and the number of features to consider (-K). Like in previous tables, all the audio features described in this chapter was used as input to the random forest classifier. 82
Table 5.19 Table showing results of multilayer perceptron classifier with different values of a variety of parameters, including the number of hidden layers and parameters for the backpropogation algorithm. 83
Table 5.20 Table of MFCC results with different window sizes using the LIBLINEAR classifier. In this and subsequent tables, "ws" refers to the size of the FFT window in samples, "hp" refers the hop size between sequent FFT frames in samples, and "mem" refers to the size of the texture window in frames. Longer texture window sizes have been omitted as they showed anomalous behaviour due to the short size of the clips as compared to the long integration time. 85
Table 5.21 Table of MFCC results with different numbers of MFCC coefficients using the LIBLINEAR classifier and the ORCACALL dataset 87

Table 5.22 A table showing the results with the ORCACALL dataset of doing a parameter search over a wide number of values of C and using the different solvers in the LIBLINEAR package. 88
Table 5.23 Table showing the classification accuracy and timing when using different FFT window sizes on MFCC features using a linear SVM kernel using the ORCACALL dataset. 89
Table 5.24 Table using audio from the ORCACALL dataset showing the effect of window size on classification performance with LIBLINEAR using the Root Mean Square (RMS) energy of a signal as a feature. 89
Table 5.25 Table showing the effect of using combinations of different statistical measures of the spectrum of a signal on classification performance with LIBLINEAR. 90
Table 5.26 Table showing the classification performance of LIBLINEAR with different combinations of Chroma, Spectral Crest Factor (SCF) and Spectral Flatness Measure (SFM). 91
Table 5.27 Table showing the result of using the YIN pitch estimator as a feature for input to the LIBLINEAR SVM classifier with different window sizes and hop sizes. 91
Table 5.28 Table showing the use of all audio features described above as input to the LIBLINEAR package. In this table "css" refers to the combination of Chroma, SCF and SFM features. 92
Table 5.29 Results showing the effect of changing the values of coef0 and gamma with LibSVM when using a sigmoid kernel on the ORCACALL dataset. 93
Table 5.30 Table showing the results of using the Radial Basis Function kernel under LibSVM using different values of the Cost and Gamma parameters. 94
Table 5.31 Table showing results of Logistic Regression classifier in the Weka software package with different values of the ridge parameters for the ridge in the log-likelihood function. 94
Table 5.32 Table showing results of Naive Bayes classifier in the Weka package with the -D parameter which corresponds to the use of supervised discretization to process numeric attribute. 94

Table 5.33 Table showing results of J48 decision tree classifier with different values of all adjustable parameters and a combination of all audio features. In this table, the parameter "-A" enables Laplace smoothing for predicted probabilities, "-C" sets the onfidence threshold for pruning, "-L" turns on functionality to not clean up after the tree has been built, "-M" sets the minimum number of instances per leaf, "-N" specifies the number of folds for reduced error pruning, "-R" turns on reduced error pruning, "-S" disables subtree raising, and "-U" uses an unpruned tree. 95
Table 5.34 Table showing results of random forest classifier with different values of the number of trees to build (-I) and the number of features to consider (-K). Like in previous tables, all the audio features described in this chapter was used as input to the random forest classifier. 96
Table 5.35 Table showing results of multilayer perceptron classifier with different values of a variety of parameters, including the number of hidden layers and parameters for the backpropogation algorithm. 97
Table 5.36 Performance results of timing on subsets of the entire Orchive dataset. 98
Table 5.37 Global accuracy for different methods of converting a Yin pitch contour into an alphabet using FTSQ 105
Table 5.38 Global accuracy for different call types using Octave Removal NonLinear 1200 parameters with the FTSQ algorithm. 105
Table 5.39 110
Table 5.40 A table showing the average number of correct annotations assigned by different populations of users. 113
Table 5.41 114

Table 10.1 Results from using different machine learning classifier algorithms on the problem of bird song classification. The accuracy of each classifier in percent correct classification is shown. 166
Table 10.2 A confusion matrix for classifing bird song using a Support Vector Machine classifier. The vertical axis describes the ground truth label for each species, and the horizontal axis describes what label each clip was classified as. 166

List of Figures

1.1 Shown is the increase in hard drive capacity from 1980 to 2010. It should be noted that the y-axis is shown in a logarithmic scale. Image from Wikipedia. 5

2.1 Alarm calls of Lawrence's, Lesser, and American Goldfinches (*Carduelis lawrencei*, *Carduelis psaltria* and *Carduelis tristis*. From Coutlee 1971, Animal Behavior 19:559. 17

2.2 The calls of the Varied Thrush *(Ixoreus naevius)*. From Martin 1970, Condor 72:453. 18

3.1 A small annotated section of audio from the Orchive 40

3.2 System Diagram 46

5.1 A graphical representation of window size, hop size and memory size in the Audio Feature Extraction algorithm that has been used. . . 60

5.2 F0 contour for 21 examples of the N04 call. 99

5.3 F0 Contour for 10 examples of N05 call. 99

5.4 Raw audio signal of Orca vocalization N01. 100

5.5 Frequency spectrum of Orca vocalizations N01 and N09. 101

5.6 Frequency spectrum of Orca vocalizations N01 and N09 102

5.7 Spectrogram, pitch and RMS view of a small portion of the Orchive catalog viewed with the OpenMIR interface. 103

5.8 Thresholded and down-sampled F_0 signal of a N01 call. 104

5.9 Octave errors within a F_0 signal of a N01 call. 106

5.10 Distribution of F_0 frequencies of call catalog. 107

5.11 Log-normal distribution for quantization 108

5.12 F_0 signal of N01 call quantized to a finite alphabet 109

5.13 A figure showing the number of visitors per day to the OrcaGame website. Four distinct peaks can be seen in this data, which correspond to the four different campaigns undertaken to recruit visitors. What is noticeable in this graph is that there was a long lasting tail of engagment of the game long after the initial recruitment pushes were done. This long tail shows that this game was very engaging for a small subset of users, and even 5 months after the study, annotations are still being made each week by volunteers. 117

5.14 A histrogram showing the frequency of repeat visitors to the OrcaGame website. This histrogram shows that the majority of visitors only came to the site once, but that a substantial number of visitors (17) came back to the size between 15 and 25 times, which shows that for a certain subset of the population studied, this simple game was very engaging. 118

5.15 A histogram showing the length of time that visitors spent on the site. This shows that a large number of users (588) only briefly visited the site, but a substantial number (22) visited the site for longer than 30 minutes. This shows that certain participants found even this very simple version of game quite engaging. 119

5.16 A screenshot of the interface by which researchers can construct new custom levels of the orcagame using a web browser. All the annotations in the database are able to be added to the interface by the researcher, who can quickly build new levels to test the skill level of users. 120

5.17 A histogram showing the average percent correct for the different communities of users recruited to play the OrcaGame. In this histogram, the label "ex" refers to experts in orca vocalizations, "st" refers to students, "fb" to facebook, "tw" to twitter, "go" for Google+, "ol" to people recruited on the OrcaLive website and "ga" for users recruited from Google Ads. 121

5.18 A histogram showing the total number of annotations provided by different populations of users recruited to play the OrcaGame. In this histogram, the label "ex" refers to experts in orca vocalizations, "st" refers to students, "fb" to facebook, "tw" to twitter, "go" for Google+, "ol" to people recruited on the OrcaLive website and "ga" for users recruited from Google Ads. 122
5.19 A graph showing the total number of annotations per user, sorted by the amount of annotations per user. What one can see is a typical long-tail type graph, where a few users contribute many annotations, and most users provide only a few annotations. 123
5.20 A screenshot of the instructions from the OrcaGame. The original form of instructions had three separate screens, and this new screen was created from suggestions from members of the pilot study. . . 124
5.21 A screenshot of the OrcaGame interface, showing the query call at the top of the screen and the the reference calls below. the screen th 125

10.1 A diagram showing the progression of data from the level where it is collected at the bottom up through derived knowledge and finally leading to public policy at the top. 162
10.2 A diagram showing the proscribed manner at which the 9 different data points are recorded for each site in the ABMI data collection protocol. 163
10.3 An image of the CZM microphone, used to efficiently collect a stereophonic field of sound but with the ability to amplify distant sounds. 164
10.4 A schematic view of the CZM microphone, used by the ABMI to efficently capture the sounds of birds in field recordings. 164
10.5 A top down view of the CZM microphone, showing the way that sound is amplified simply with the physical design of the microphone. 165

ACKNOWLEDGMENTS

I would like to thank:

my parents, Fern and Randy Ness for their unfailing love and support,

NSERC, for funding me with a 3 year Ph.D. Fellowship

The world is transitory. You will find stability only on the path of Karma Yoga. Only action can take a man to God and give him liberation. ... Brave ones, all of you, continue to work! Through Karma alone will you be able to change the world. It is the only way.

Babaji

DEDICATION

To my parents, Fern and Randolph Ness.

Chapter 1

Introduction

In recent years, advances in computerized recording, storage and processing technology have enabled researchers in the Biological Sciences to collect, digitize and store very large archives of bioacoustic data from a wide variety of species. The size and number of these large bioacoustic archives is growing rapidly, and the creation of tools to help biologists make sense of this data is an area of ongoing research [Hal13].

Concurrently in the field of Computer Science, advances in audio feature extraction and machine learning have made it possible to extract meaning from raw audio, as is shown by the successes of the field of Music Information Retrieval [Tza08]. In our lab, we are interested in studying these large bioacoustic archives with new audio feature extraction and machine learning algorithms. However, for each of these large bioacoustic archives, considerable knowledge of the vocalizations of the species represented in the archives is required. For example, in order to annotate a field recording of bird songs, knowledge of the different vocalizations of each of the birds present is required, and experts in birds vocalizations from one region might not be experts in another region. On the other hand, the rapid pace of development of audio feature extraction and machine learning algorithms would make it difficult for a biologist to be an expert in the currently accepted most effective way to use them.

What is required is some way for computer scientists and biologists to collaborate so that the extensive biological knowledge of the biologists can be combined with the tools and knowledge of computer scientists. Many successful examples of this collaboration have occurred in the past [Yur05] with smaller datasets, with biologists using tools that run on such as Raven [oO11] to annotate recordings, and then either using algorithms themselves [VDS99] or collaborating with computer scientists to extract information from these recordings.

The vast size of the new bioacoustic archives presents challenges for these traditional tools based on a single computer model of computation. The first problem is simply the vast size of these audio collections, with a dataset in the range of tens of terabytes [NSST13] to petabytes [UCKP09], it would be impractical for each computer to have its own copy of the data. Furthermore, it would take an impractical amount of time to extract audio and use machine learning on these datasets on a single computer.

For many of these sources of bioacoustic data, there are only a very few number of experts that are capable of annotating this data, and because they often spend considerable time in the field, the time they have to annotate recordings is very limited. It should also be noted that the biologists who have the most knowledge of the vocalizations of the species of interest are often are not directly working on the same project that the computer scientists are working on and would only indirectly obtain benefits from doing the large number of annotations that are required. In some cases, such as the Right Whale project from Cornell university or the Alberta Biodiversity Monitoring Institute, biologists are hired specifically to annotate recordings. In the case of the Orchive, most of the annotations were graciously volunteered by biologists working on other projects, and the labels did not directly benefit their research. It was difficult to obtain annotations from more experienced orca researchers simply because they were very busy on their own research projects.

However, some of these bioacoustic archives are of quite charismatic species, such as whales, birds and frogs, and there are many members of the public that already listen to recordings of these species for pleasure. There already exist systems for enabling these members of the public to be citizen scientists and to help biologists by annotating data. With sufficient training and monitoring of data quality, the contributions of these citizen scientists can be considerable.

In this thesis, a system is developed and presented to help biologists and computer scientists collaborate on the annotation, segmentation and classification of large bioacoustic archives. This system uses web-based technology to allows groups of biologists to listen to, view and annotate recordings, and enables computer scientists to extract audio features from these recordings and to use the annotations by biologists to train machine learning systems. This system also has the functionality to display the output of the audio feature extraction and machine learning algorithms in a form that both the biologists and computer scientists can easily use. This system also provides a serious casual game interface that allows citizen scientists to help annotate data,

and allows both the biologists and computer scientists to use these annotations to derive knowledge from the archive.

The ultimate goal of the use of this system is to take a large unannotated bioacoustic archive and to segment and annotate it using a combination of expert knowledge, annotations from citizen scientists and to classify its audio features using machine learning systems. The system we present will use data from The Orchive, a large collection of over 20,000 hours of orca vocalizations collected over the last 30 years by OrcaLab. This system is also amenable to use on different bioacoustic archives, and in Appendix A preliminary investigations on a large archive of the recordings of birds from the Alberta Biodiversity Monitoring Institute will be presented.

This system will be evaluated in three different ways. The first and most important will be to measure the classification accuracy of the machine learning system. The second will be to measure the classification accuracy and engagement of citizen scientists recruited from different communities using the serious casual game interface. The third will be to investigate the engagement and effectiveness of this system as used by biologists to annotate recordings. Finally, conclusions will be presented about the effectiveness and practicality of these different techniques to annotate large bioacoustic archives.

1.1 Contributions

This thesis will demonstrate the following novel contributions to the field:

1) The development of a web-based interface that allows for experts in bioacoustics to upload, view, listen to and annotate recordings.

2) Development of a system to allow researchers to quickly and easily build versions of a simple casual game based on a matching paradigm that they can deploy and collect data from citizen scientists in order to help annotate large databases. Results are presented using a variety of different populations of users, including in-person tests, expert users, people of the OrcaLab community, tests using undergraduate and graduate students via an emailing list, social distribution of the game using Facebook, Google+ and twitter, and users recruited through the use of Google ads.

3) Development of a system that integrates a number of different packages for extracting audio features from recordings and to display those features to users. This system is highly interactive and allows them to quickly change parameters of the algorithms and view the data using a web-based interface.

4) Development of a system to allow for researchers in bioacoustics to quickly and easily generate training and testing sets of data from recordings, to train machine learning classifiers on this data, and to run these classifiers in real-time on data.

5) Development of a system to allow researchers to run these audio feature extraction and machine learning programs on large amounts of data using clusters of computers, and to then view the results of these computations in a web based interface. This system allows for both the use of traditional grid based resources on datasets, which works well with a number of problems in bioacoustics that are embarrassingly parallel.

6) Testing of the effectiveness of different audio feature extraction and machine learning algorithms on bioacoustic data, and results from using these algorithms. This includes the use of spectral based audio features, such as Mel-Frequency Cepstral Coefficients (MFCC), autocorrelation based approaches such as the Yin pitch detection algorithm. The effectiveness of different classification algorithms using these audio features is explored, using algorithms such as Support Vector Machines, Multilayer Perceptrons, Naive Bayes and Decision Trees.

7) The development of two publically available datasets ORCAOBV and ORCACALL that contain hand curated clips containing orca/background/voice annotations in the case of ORCAOBV and 11 different calls in the ORCACALL dataset. At the recent ICML 2013 Workshop on Machine Learning for Bioacoustics workshop [Hal13], the lack of good bioacoustic datasets designed for Machine Learning researcheres was mentioned by a number of participants, and a call for new bioacoustic datasets was made. These two datasets have been made available to the Machine Learning community and are in a format readibly amenable to testing of new machine learning systems, with raw audio, labels and audio features being made available. This data can be downloaded from the following website (http://data.orchive.net).

1.2 Large Bioacoustic Archives

The storage capacity of computer hard disks has increased in almost an exponential manner since 1980, as is shown in Figure 1.1. This dramatic increase of storage capacity has made it possible for very large archives of bioacoustic data to be stored in digital format.

Many such archives that were previously stored on analogue magnetic tape have begun to be digitized and analyzed and presented to the research community and

Figure 1.1: Shown is the increase in hard drive capacity from 1980 to 2010. It should be noted that the y-axis is shown in a logarithmic scale. Image from Wikipedia.

public through online web resources. The Cornell Lab of Ornithology is one such organization, and has recently made available a huge amount of the recordings of birds through their website, the Macaulay Library [Liu07] and contains more than 175,000 audio recordings. Another project from the Alberta Biodiversity Monitoring Institute to monitor the biodiversity of birds using teams that manually record audio has been operating since 2002 and has collected approximately 8,800 individual 10 minute recordings [BHS+09], with more each year as the project ramps up, this year they collected approximately 1,800 new recordings and expect to collect increasingly more each year.

Here at the University of Victoria, we have developed The Orchive, one of the largest repositories of bioacoustic data in the world, containing over 20,000 hours of recordings of orca vocalizations, collected from a land based research station at Hanson Island on the BC coast. The Orchive project is the primary focus in this thesis, the project will be described in detail in the Case Studies chapter and results from using the system developed in this thesis to this dataset in the Evaluation

chapter.

The same advances in computer storage technology have led to researchers becoming even more adventurous in the collection of large amounts of bioacoustic data, skipping the process of recording onto analog tape and recording directly into the computer. The VENUS and NEPTUNE projects are cabled undersea observatories that continuously record many kinds of data, including salinity, pressure and video, and of relevance to this thesis, audio data. Another such project to continuously record audio data is from Cornell University, and is a project to remotely record the vocalizations of Right Whales in the Atlantic. In this project, researchers have deployed a set of 8 buoys recording audio continuously from 2008 to the present, and have collected over 100,000 hours of audio from these remote sensors [UCKP09]. Cornell University has a similar program to record the vocalizations of Blue Whales in the eastern Atlantic that has collected a comparable amount of data. There are many such projects, and more and more of them are being started over time.

The amount of audio data recorded by these various projects is truly immense, and in order for researchers to make sense of this data, tools to navigate, listen to, annotate, analyze and classify are becoming increasingly more important. Cornell University has developed such a system which allows for researchers to access a central repository of data from their workstations using MATLAB, and this project has been used to find the vocalizations of Right Whales and to monitor the behavior and population of this threatened species.

This thesis describes work in applying advanced audio feature extraction, analysis and visualization tools to the study of large archives of bioacoustic data. It focuses on the data from the Orchive, but can be used for other sources of bioacoustic data as well. There are three distinct types of tools that will be demonstrated, the first are tools to extract features and analyze audio. The second set of tools are web-based and allow users from around the world to collaboratively view and analyze the results obtained from the first set of tools. The third set are interfaces that use a casual game metaphor to allow citizen scientists to help provide annotations on this audio.

An aspect characterizing this work is the need to collaborate with domain experts in the vocalizations of the biological species of interest, and a large amount of the effort in this project is devoted to the development of web-based interfaces that allow domain experts with varying degrees of computer sophistication to access and make sense of the extracted data that our tools produce. Thus, the core part of this work is to bring together tools, data, biologists, computer scientists, and citizen scientists

together into a collaborative partnership.

This work draws on ideas and concepts from many disciplines. Because of this it is essential to include not only definitions of these concepts, but also the fields from which them come. These are presented in the Glossary chapter.

1.3 MIR and Bioacoustics

It has only been in recent years that computer hard drives and RAM has become able to store the large amounts of data that sound represents. This represents another quantum leap for the field of bioacoustics, as it allows for large amounts of audio data to be quickly accessible, and for this data to be indexed and stored in databases and analyzed by computers. The field of Music Information Retrieval experienced a similar blooming in the early 2000's, when computer storage and computational power became great enough to store and analyze the data of songs. The field of bioacoustics is just starting to show similar growth, and the use of tools from the field of Music Information Retrieval (MIR) on bioacoustic data has shown great promise.

However, many of the tools developed for MIR are not well adapted for the study of bioacoustic data. When studying recorded songs, there is often a large amount of well-curated meta-data for each song, For example, when classifying songs based on genre, the artist, song title, genre, record label and many other forms of data are available, and boundaries between songs are clearly marked, and are often in individual files. Music often come pre-segmented into songs, which often have identifiable sections including verse and chorus, as well as lower level features such as beat and tatum that facilitate analysis by computers. In addition, most work in MIR has been on songs that were professionally recorded in a studio environment.

In bioacoustics, this is often not the case, and in the cases of large bioacoustic databases, recordings are often taken from a single location, and how close animals are to the recording devices can change dramatically during a recording. There are often also many sources of other sound in the recording, from environmental noise like wind, to human produced noise like from boats or cars. Also, in many cases there are a variety of different animals making sound in a recording, and these sounds can overlap each other. While some bioacoustic recordings are well segmented, such as those of the recordings of bird songs from the Cornell Lab of Ornithology, in many cases of continuous recordings, the locations of the bioacoustic sounds are not localized in time, and these recordings must be annotated and segmented before they

can be analyzed.

In most studies of bioacoustics up to the present time, individual researchers record the sounds of the animals that they are interested in studying. In the process of doing the recording, they make notes and record other kinds of metadata about the audio they record. The amount of audio that is typically analyzed is of the range of hundreds to a few thousand recordings. Even in larger studies such as those by Harald Yurk on the vocalizations of Orcas, the dataset is of the order of 1000 recordings. These recordings are typically analyzed on a single computer using software such as Raven [oO11], a powerful tool for the study of bioacoustic data produced by the Cornell Lab of Ornithology. This software allows researchers to record, import, view, analyze and annotate recordings and provides ways to export the annotations to other programs that can be used to further analyze the audio. It works best with shorter audio files, although large files can be read in and viewed using a paging metaphor, where sections of several minutes of audio are visualized at a time.

In recent years, the larger storage and computational capacity of computers has inspired researchers to analyze larger and larger collections of bioacoustic data. Much of the historical audio recordings are present on audio tapes, and using high-throughput audio digitization facilities, this data has begun to be transferred to digital form. At the University of Victoria, we have previously described a project called The Orchive [NSST13] where we have digitized over 20,000 hours recordings from the OrcaLab research facility, stored originally on 45 minute long analog audio cassette tapes. These recordings contain large numbers of the vocalizations of Orcas (*Orcinus orca*) along with other species of marine mammals.

With the advent of computers that can store this large amount of data, even more ambitious efforts have recently begun to collect continuous data including audio. Two of these projects are the closely related VENUS and NEPTUNE ocean observatories, part of Ocean Networks Canada. In these projects, permanent undersea observatories that are directly connected via cable links to land collect continuous data, including echo sound data, video and audio, as well physical parameters such as salinity, pressure and depth readings. This data streams in from 6 different observatories, and in the case of audio data, in many of the locations, multiple channels of high sample rate audio are collected. This project has already collected thousands of hours of audio, and is planned to collect data for the next 25 years.

The advent of very large bioacoustic databases of tens or hundreds of thousands of hours presents problems to currently existing tools. Even in the case of the 45 and

30 minute recordings from the Orchive and ABMI projects, current tools have some issues with allowing researchers to quickly load the audio, view it, and quickly navigate around within and between these files. In addition, the storage space required for these large audio collections makes it impractical for researchers to have the whole collection on a local computer. In the case of the Orchive, the entire dataset takes about 20TB of disk space when stored as uncompressed audio files.

The Music Information Retrieval tools used to analyze this audio are also impractical to use on a single computer, the time it takes to analyze the audio, and the amount of storage space makes it impractical to analyze large collections of audio data on a single computer. It is also important for researchers to be able to collaborate on these large scale projects, to share their annotations, audio data, and the raw results of their analysis with colleagues. In order to do this, one possible approach would be to use a web-based system, where the individual researcher can connect to a large server-based system that presents the data to them in an easy to use form, allows them to make and share annotations, and connects to large amounts of computing resources for them to perform audio feature extraction, machine learning and other forms of analysis on their data.

Web-based software has been helping connect communities of researchers since its inception. Recently, advances in software and in computer power have dramatically widened its possible applications to include a wide variety of multimedia content. These advances have been primarily in the business community, and the tools developed are just starting to be used by academics. We have been working on applying these technologies to ongoing collaborative projects that we are involved in [NWMT08]. By leveraging several new technologies including HTML5/Javascript, Node.js and Python, we have been able to rapidly develop web-based tools. Rapid prototyping and iterative development have been key elements of our collaborative strategy. Although our number of users is limited compared to other areas of multimedia analysis and retrieval, this is to some degree compensated by their passion and willingness to work closely with us in developing these tools.

1.3.1 Relevance of this work

An important aspect in the design of a tool to support collaborative work is to consider what user communities will use the tool. In the case of the Orchive, there are a number of different scientific communities that will be using this tool and the data this tool

provides access to. The primary scientific community that will benefit from this work will be cetacean biologists. In order to study the rich archive orca vocalizations that have been recorded by OrcaLab, researchers must currently travel to Hanson Island, search through the lab books and incidence reports to find which recordings contain the data they are interested in, locate the physical cassette tape corresponding to this recording, and then either manually listen to the tape, or perhaps digitize the tape and analyze it in the computer. Each researcher typically then keeps the annotations and data generated from this procedure themselves. If future researchers want to obtain this data for further analysis, they must first be aware of the fact that this researcher has the data, and then request it from them. With the distributed collaborative system we have designed, not only can these biologists easily listen to any recording in the entire archive from any internet connected computer in the world, and compare different recordings, they can also add their annotations to the system. These annotations can be either private or public. If they are for use in a publication, after the article has been accepted for publication, the researcher can make their private annotations public. These researchers are less interested in the details of audio feature extraction and machine learning algorithms, and are instead more focused on asking biologically informed questions, like dialect change in cetacean call repertoire [DFS00].

Another scientific community that will receive benefits from this archive are the developers of bioacoustic algorithms. These scientists are typically computer scientists with interests in Music Information Retrieval and bioacoustics. This archive represents a site where researchers can get large amounts of high quality and uniformly collected data. Researchers interested in bioacoustic algorithms have different goals and skill sets from cetacean biologists, for example, many have extensive knowledge of Digital Signal Processing and audio feature extraction algorithms. This system should be flexible and powerful enough to allow these researchers to ask questions that are relevant to them. The required features for this group of users include allowing them to choose different audio feature extraction algorithms, and to then take the resulting data and run it against a variety of machine learning algorithms in as flexible a manner as possible.

Another group of scientists that have expressed interest in the Orchive are Environmental and Conservation scientists. A research question of particular interest is the effect of boat noise [FOH04] on cetaceans and on the marine environment in general. For these researchers, the data they will be most interested in is the frequency

and nature of orca vocalizations, and the intensity and spectral characteristics of boat noise [HNV+09]. There are large differences in the intensity and frequency content of boat noise depending on the type of boat that creates it, speed pleasure craft often create a high pitched noise that quickly moves away, tug boats have a lower pitched sound and take a long time to move through an area, and cruise ships make a loud and distinctively high pitched sound. Analyzing the effects of these various types of boat noise will help researchers to establish guidelines for boat noise as it affects this sensitive population of marine mammals [DRGOH+09].

Another group of scientists that this work will benefit are those studying the social organization of whale communities [BOE+90]. There have been studies that investigate the transmission of culture in orca societies [DFS00] and have found evidence of this through the examination of dialect change. In a similar vein, other studies have investigated social learning [JS00] in communities of orcas. With a large database such as this, more of these type of studies will be made possible in the future.

Another large and difficult problem domain that requires the skills of researchers from many different fields is the investigation of orca vocalizations. The whale species *Orcinus orca* is found around the world, in places as far afield as Antarctica and Iceland. There are three distinct types of Orcas, Transients, Residents and Offshores, each of which have different feeding behaviors and different styles of communication. The vocalizations of orcas are complex and diverse, and consist of a wide variety of vocalizations, which include echolocation clicks, tonal whistles and pulsed calls [DFS00].

Around Vancouver Island there are two distinct communities of Orcas, the Northern Residents, which have a range north of Campbell River, and the Southern Residents, which spend time around Victoria. Orcalab is a research station located on Hanson Island, a small island up near the top of Vancouver Island, directed by Dr. Paul Spong and Helena Symonds and which studies the Northern Resident population of Orcas. Orcalab has been recording Orca song for over 20 years, and has amassed a huge archive of over 20,000 hours of audio on magnetic cassette tapes. In a collaboration with them, we are digitizing and analyzing this huge and rich archive in a project we have called The Orchive.

The goal of this thesis is to create a system that enables distributed cognition between these three facets, allowing expert users to collaborate, giving them advanced machine learning algorithms to help them analyze data, and in cases where datasets are enormous, using the power of crowdsourcing to pore through large datasets. In

this thesis we describe an application of this approach to the problem of analyzing the orca vocalizations.

The main motivation behind our work has been creating better interfaces for interacting with the Orchive [TLSS07] a large archive of audio recordings of Orca vocalizations. There are stable resident populations of *Orcinus orca* in the northwest Pacific Ocean, and some of these populations [JFB00] are found near Hanson Island, off the north tip of Vancouver Island in Canada. Orcalab is a research station that has been recording audio of these Orca populations since 1972 [VDS99]. They have amassed a huge archive of more than 20,000 hours of audio recordings collected via a permanent installation of underwater hydrophones.

Most of the existing work in the automatic analysis of Orca calls has focused on detection and classification rather than retrieval. A real-time system with low computational requirements for the detection of Orca vocalizations is described in [LMHFJ10]. Annotation bootstrapping is a technique used to classify/segment hydrophone recordings into three broad categories: voiceover, background, and vocalizations [NWMT08].

Our work was influenced by two publications that described different representations and methodologies for the classification of Orca calls. Orca vocalizations consists of well-defined, discrete calls with tonal signal components. They can be characterized by the pulse rate contour of the goal, which can be viewed as analogous to the pitch contour of a speech or monophonic music signal. A method for computing acoustic similarity between pulse rate contours (normalized so that they all have the same duration) using the discriminative error of a Artificial Neural Network is described in [VDS99].

1.4 Scope of this work

The fields of Music Information Retrieval (MIR), Visualization and web-based Human Computer Interaction (HCI) are each vast topics in their own right, not to mention the application areas of Ethnomusicology and bioacoustics. So as to make the present work tractable, we will focus on three very specific problems and will apply a carefully selected subset of some of the tools in MIR, Visualization and HCI to help us in developing solutions for these areas.

In the field of the analysis of bioacoustic signals from *Orcinus Orca* vocalizations, we will use tools from MIR, including Fast Fourier Transforms (FFT) to analyze and

display spectrograms, along with tools such as Mel-Frequency Cepstral Coefficients (MFCC), average zero crossing rate, and spectral centroid, along with many other such tools, to help us visualize and analyze orca vocalizations.

In terms of Machine Learning algorithms, this thesis will concentrate on supervised learning algorithms. In these algorithms, data is labeled according to a user defined dictionary and the Machine Learning algorithm is trained to recognize the features in this data and to predict the correct labels of new test data based on an internally constructed model of the training data. This is in contrast to unsupervised Machine Learning algorithms which do not require label-led data and perform tasks like clustering or association analysis.

Chapter 2

Related Work

This thesis explores different approaches to allow scientists to view, annotate and analyze large bioacoustic databases. The study of bioacoustics has a long and rich history, and in the first section of this chapter, a brief summary of the history of bioacoustics in general will be presented. Previous work done in analyzing the vocalizations of the species *Orcinus orca* will be reviewed.

In the following section, an overview of the literature on the use of audio feature extraction to generate features suitable both for the visualization of bioacoustic data, and for the use of machine learning algorithms will be presented. A brief history of the use of waveform-based methods for visualization of sounds will be provided, and this will be followed by reviews of the use of spectral based methods, including sonograms, spectrograms and measures derived from spectrograms, including Mel-Frequency Cepstral Coefficients and other descriptions of spectra. A brief overview of the field of pitch detection, in which algorithms that primarily use the autocorrelation of a time based representation of sound and of the use of these methods in the study of bioacoustics will be provided.

In order to allow scientists to listen to, view and perform analysis on these large collections of audio, the use of computer based tools is a necessity. In the past, researchers in bioacoustics would analyze their data with programs on a single computer such as Raven and MATLAB. We will provide only a brief search of the vast literature on the use of computer based tools to view, annotate and analyze audio. In ideal situations, these tools provide for Intelligence Augmentation (IA), in which computer based tools do not replace human intelligence but instead augment or improve it. This stands in contrast to the concept of Artificial Intelligence, where tools that produce information independently of humans are used, with the ultimate end of

replacing the need for human intelligence. IA instead acknowledges that computers and humans have different cognitive strengths, and that for certain complex tasks, an optimal system would combine the best aspects of human and computer intelligence. We will examine the literature for evidence of tasks that benefited from the use of IA, and will draw parallels between these systems and the system described in this thesis that allows scientists to view, annotate and analyze large bioacoustic databases.

One of the largest hurdles in the study of large bioacoustic databases is their vast size, the small number of researchers studying these databases, and the messy nature of these databases. In order to assist scientists with large datasets, the use of citizen science and crowdsourcing has been used for many years, with approaches as diverse as the collection of word use for the Oxford English Dictionary, the use of far-flung citizen scientists to aid Linnaeus with the study of taxonomy, to the work of Dr. Fred Urquhart in the study of monarch butterfly migrations. In recent years, the use of the internet has enabled many more ways for interested members of the public to help scientists study difficult problems, and we will provide a brief summary of the literature on this topic. In this thesis, we present a simple matching game to allow citizen scientists to assist us in the classification of audio, in a paradigm we call Serious Casual Games. We will examine the literature on the use of gamification on the study of complex phenomena and what aspects of gamification have shown the most promise in previous studies.

2.1 Bioacoustics

Humans have used sound to identify animals from time immemorial, some of the earliest extant literature comes from Aristotle. In On the Soul [Arib] he describes how animals make sound, and notes that while some sounds are produced by the "impact of the inbreathed air against the windpipe", there are many other mechanisms by which animals produce sound. In The History of Animals [Aria], he describes a wide variety of sounds made by animals, from the "low hiss" of the tortoise, the "peculiar croak" of the frog, to the "squeak and moans" of a dolphin.

Animals and insects make a wide variety of sounds, and humans have been interested in and studied these sounds since the time of Aristotle and before. The sounds made by animals are produced for a wide variety of purposes, from the echolocation clicks of bats, to alerting others to the presence of threats, to communication for finding mates, and for other kinds social interaction. These sounds are often quite

characteristic of the species in question, and humans have long been able to identify different animals by the sounds that they make.

The sounds produced by animals and insects can be produced in a wide variety of ways, and can be of a wide range of frequencies and amplitudes. The range of human hearing is approximately 20 Hz to 20,000 Hz, [Lyo90] and while many animals make sound in this range, others, such as bats, make sound in the ultrasonic range, while others like elephants, produce sound below the range of human hearing. The amplitude of sounds produced by animals is similarly diverse, with some producing sounds inaudible to humans, while others like the blue whale can produce sounds in excess of 188 decibels [CA06].

Before the advent of electro-mechanical and electronic recording devices, sounds were often described by researchers using transliterations of animal sounds onto human speech, a simple example is the song of the Chickadee *(Poecile spp.)* which is represented in human speech with the words "chick-a-dee-dee" or the sound of the Tawny Owl *(Strix aluco)* which is represented by the words "tu-whit tu-whoo". An interesting historical aside is that in different languages, different words are used to represent the sounds of animals, such as the "meow" of a cat in English, "miaou" in French, "miyav" in Turkish and "nyan nyan" in Japanese.

The earliest quantitative study of animal sounds began with the recording of sounds onto wax cylinders, and the visualization of sounds was a byproduct of the way that sound was recorded onto these cylinders. The creation of permanent recordings of animal sounds was a huge leap forward for the field of bioacoustics as it allowed researchers to record the sounds of animals and to listen to and analyze them in more detail in a laboratory environment. It also allowed the scientists to distribute their recordings to other researchers, and facilitated scientific discourse in this field.

Early devices such as the Phonoautograph of Edouard-Lyon Scott de Martinville developed in 1857 and the Phonodiek of Dayton C. Miller developed in 1909 allowed for the visualization of the waveforms of sounds, and Alexander Graham Bell created a device that permanently recorded traces of sounds onto smoked-glass. It is relatively difficult to analyze bioacoustic sounds purely by their waveforms, as they are often composed of multiple overlapping harmonics, and the relative phase of these harmonics can make sounds that sound similar to the ear, but look quite different when visualized as a waveform.

Bioacoustics is a scientific field of study that combines the fields of biology and acoustics. Although humans have used the sounds of animals to identify and track

Figure 2.1: Alarm calls of Lawrence's, Lesser, and American Goldfinches (*Carduelis lawrencei*, *Carduelis psaltria* and *Carduelis tristis*. From Coutlee 1971, Animal Behavior 19:559.

them for many years, one of the first researchers in this field was Ivan Regen who in 1925 systematically studied the sounds of insects [Zar29]. In the later half of the 20th century, advances in electronic means of recording and producing sound dramatically increased the breadth and scope of the field of bioacoustics. In recent years, the application of the computational tools used in Music Information Retrieval have further extended the possibilities of analyzing sound from biological sources.

Because of the sensitivity of humans and other animals to the frequencies of sounds, devices that could separate sounds into their characteristic frequency bands proved very useful in the study of bioacoustics sounds. The first such commercially available device was the Sona-Graph, produced by the Kay Electric company. This device produced a two-dimensional representation of sound, with frequency along one axis and time along the other. It had two settings for the production of images, a narrow-band mode and a wide-band mode. The narrow-band mode had good frequency resolution at the expense of time resolution, an example of this is shown in Figure 2.1. The wide band mode emphasized time resolution at the expense of frequency resolution, an example of this is shown in Figure 2.2. The Sona-Graph was widely used in the field of bioacoustics and was used to describe the songs of many birds and other animals and insects.

While originally bioacoustic data was recorded onto wax cylinders, for most of the 20th century, audio was recorded onto analog magnetic tapes. These tapes were capable of storing large amounts of data, and allowed researchers to more easily record

Figure 2.2: The calls of the Varied Thrush *(Ixoreus naevius)*. From Martin 1970, Condor 72:453.

the sounds of animals in the field and the lab. Analog audio tapes were supplemented by the introduction of Digital Audio Tapes (DAT) by Sony in 1987. However, one limitation of both of these forms of storage is the long time it takes to seek to a particular place on the tape, due to the linear way that they store data on tape.

2.1.1 Orcas

There is a considerable, but not immense, literature on the study of the bioacoustics of *Orcinus orca*, also known as the Killer Whale. One of the earliest studies was described by a paper in 1967 by Singleton and Poulter [SP67] in which a detailed spectral analysis of the calls of captive Killer Whales in which a digital approach using a Fast Fourier Transform was used to produce an spectrogram of short regions of audio. This study examined pulsed calls, and made reference to work by Watkins [Wat67] in which an analysis of pulsed calls was made. These pulsed calls were identified by Watkins to be short pulses of tones, and he provided a detailed analysis of the spectral qualities of these tones, noting that when studying pulsed calls, the nature of the pulses has a strong impact on the spectrogram, and gives rise to a banded structure, and that the distance between the bands gives the repeat rate of the pulses, this is also known as the Side Band Interval (SBI). In later work, especially the catalog of orca vocalizations by Ford [For87], the SBI is one of the primary audio features used in classifying orca calls. However, in the case of The Orchive, due to the extreme distance of an orca from the hydrophone this harmonic structure is often

not seen, which limits the use of the SBI in this work.

The structure of the sound producing organs in odontocetes in general was described in detail by Cranford [CAN96] who found that the tissues responsible included a structure that resembled lips and was made of connective tissue, a cartilaginous blade, a stout ligament and an array of air sacs made of soft tissues. This sound producing organ is capable of producing two sounds at once in a process known as biphonation. They noted that the sounds that could be produced included clicks, whistles, and a sound that was the combination of both, a pulsed call that could contain both a Lower Frequency Component (LFC) and Upper Frequency Component (UFC). In later papers, [Cra00] [W06], they described this sound producing organ in more detail. Brown [JC.08] did a study of the mathematical implications of this pulsed biphonation, extending the earlier results of Watkins [Wat67]

A paper in 1982 by Dahlheim and Awbrey [DA82] provided a classification of the sounds of captive killer whales, these whales had been taken from British Columbia, and a number were identified to be Northern Resident Killer Whales (NRKW), which makes it very relevant to this thesis. In this paper, they describe 11 different categories of sounds, these are upscreams, downscreams, creaks, whines, whistles, tones, buzzes, ricochets, click bursts, chatter and seesaw. In this paper they note that from the different vocalizations, they are able to classify individuals by oceanarium with high accuracy.

An extremely detailed investigation of the vocalizations of the killer whales off the coast of British Columbia was carried out in a series of papers by John Ford [BD82] [FF83] that described the group-specific dialects this population. In 1987 Dr. Ford published a large catalog of the pulsed calls of the Northern Resident Orcas [For87]. In this hugely important resource, he used a systematic nomenclature to describe a catalog of approximately 78 different calls with 24 different subtypes. The calls were labeled starting with the letter "N" for the Northern Residents, in contrast to calls from the Southern Residents which were prefixed with the letter "S". The calls were numbered according to the time that they were first encountered, starting with N1. The recordings were made into Sonagrams using a Kay Electric company sonograph, and the Side Band Intervals were measured with an electronic digitizing tablet. This call catalog remains the primary resource for researchers who wish to study orca calls, and it was hugely relevant to the research in this thesis.

The calls in this call catalog were examined in more detail in a paper in 1989 by Ford [For89]. In this paper, the vocalizations of NRKW are divided into three

categories, “clicks”, “whistles” and “pulsed calls”. The clicks were short pulses of sound that were typically made in series, with a variable duration of between 0.1ms and 25ms, and could range in repetition rate from a few pulses per second to over 300 per second. These clicks were identified to be primarily of use in echolocation. Orcas do considerable amounts of echolocation to navigate, locate other group members and locate prey [BLFH96]. Whistles were the second type of sound, and are a single narrow band tone with little or no harmonic structure. He noted that these whistles were between 1.5kHz and 18kHz and ranged in duration from 50ms to 12 seconds. These whistles were of widely varying character and did not fall into distinct structural categories.

The third type of call is the pulsed calls, and these are the primary calls of interest both in the work of Ford and others, and also of this thesis. They are of a pulsed nature as described above, that is, they have a central tone that is rapidly pulsed with abrupt and pattern shifts between pulse rates. Pulses typically had repetition rates of between 250-2000Hz, and a primary energy of between 1 and 6kHz. He noted three categories of pulsed calls, discrete, variable and aberrant calls. Discrete calls had distinctive structural characteristics and were repetitive, variable calls were modified versions of these discrete calls, and aberrant calls were clearly based on discrete calls, but were heavily modified. In this paper, he studied the amount of these three classes of calls in different behavioral circumstances, that of foraging, traveling, resting, socializing and beach rubbing, and found that more variable and aberrant calls occurred during socializing and beach-rubbing. He also did a transition state matrix of the different call types and found clear evidence that certain calls were more likely to be followed by other calls. The most striking of these was the N7/N8 pair of calls, in which an N8 call was always preceded by an N7 call. This paper is very relevant to this thesis in that it presents a theoretical structure for classifying orca calls and for the study of their vocalizations. Although it describes results from a large amount of data, the number of recordings in the Orchive is even larger, and testing the hypotheses stated in this paper with a larger dataset could be enlightening in the study of orca calls.

Orcas from different parts in the world also have discrete calls that are diverse, and a study by Parijs [PLS04] on Norwegian Killer Whales found that none of the calls recorded matched those with those of a previous call catalog [Str95]. Discrete call repertoires are thought to be mainly to promote group cohesion and to coordinate intragroup activities [For89] [For91]. Dolphin signature whistles function as cohesion

calls when a social group is separated [JS98] [WOTW05]. Resident orcas increase use of family specific calls after calves are born [WLSS06] particularly between mothers and their dependent offspring.

A recent study [PMS04] conducted with Northern Resident Killer whales found that there was evidence for call type matching in vocal exchanges. Further work by this same group [Mil06] found that discrete calls are at an amplitude that far exceeds the intra-group separation of members, and likely function in inter-group communication. Work by this group found that individual orcas tend to match call types produced by other group members [PMS04] and this can assist group cohesion [Mil02] and allow individual recognition [MSP07].

A study using data from OrcaLab, but before the Orchive was created [WLSS06], found that orcas change their call behavior in the presence of other groups of whales increasing the number of matriline-specific, variable and aberrant calls, and decreasing the number of low arousal calls. The calls also changed more when a different subclan was in the area, and might play a role in coordination of inter-group activity.

Janik [Jan99] did a study on the classification of dolphin whistles using humans versus an algorithmic approach and found a greater probability of humans to identify signature whistles than algorithmic based approaches. This is highly relevant to this thesis in that it highlights the importance of using external validation in studying vocalizations, and the serious casual game interface allows researchers to easily create new experiments for humans to identify calls. These results can then be compared with algorithmic based approaches to validate their performance. Recent work by Janik and King et al. [KSW$^+$13] suggests that the use of vocal copying plays an an important role in the social behavior of dolphins, and that dolphins can learn the vocalizations of other dolphins.

In a recent paper by Yurk et al. [YFM$^+$10] they note that due to vocal difference between clans, acoustic monitoring is a good way to monitor populations of orcas, and identified 7 different pods of whales. There have been other studies of marine mammals with fixed hydrophones including humpback whales [FMB99], bowhead whales [CH85], blue whales [SFC98] and killer whales [MS02]. In this study, the hydrophone was manned by a single observer (Michael Brittian) who did the recordings of the different pods. In this thesis we describe a system that could be used for the automated detection and recording of orca calls, thus allowing more sites to be observed and the reduction of human effort.

However, most of the work in analyzing orca vocalizations has been painstak-

ingly done by hand, sometimes using audio cassettes and expert knowledge of orca vocalizations, and sometimes by digitizing vocalizations into a computer and then analyzing spectrograms. It would be of advantage to the field to be able to apply these algorithms to a large dataset and provide collaborative visualization tools to explore it.

2.1.2 Orcas - Algorithms

For a number of years, computers have been applied to the study of orca vocalizations. One early work [VDS99] used neural networks to determine similarity between orca vocalizations. In this paper, they found that neural networks were able to predict similarity between calls with an accuracy comparable to that of expert human listeners. Another algorithm that has been applied to this problem domain is that of Dynamic Time Warping [JM06] [BM07]. In these two papers, Brown et al. investigate the use of DTW for calculating similarity between two orca vocalizations and find that this algorithm performs very well. DTW is used frequently in the field of MIR, and this thesis discusses its use as applied to chant traditions from cultures around the world in another chapter.

The mathematical foundations of the pulsed vocalizations by orcas have also been studied [JC.08]. In this paper, formulas for their spectra are rigorously derived from the basic formulas of Fourier analysis. This paper describes in detail the complex spectra that are able to produced by orcas and the biological foundations that underlie them.

Another mathematical method that has been used to describe orca vocalizations is the Hilbert-Huang transform, which has been shown [Ada06] to have advantages over using traditional Fourier based methods of calculating spectra.

One important pitfall in the automated classification of orca vocalizations [DJ06] is that there exist non-linearities in sound perception in all animals, including orcas[NRH$^+$99]. When designing tools and algorithms to study these vocalizations, it is important to recognize the differences between human and orca perception of sound.

By definition, orca pulsed calls contain a Low Frequency Component (LFC) with fundamental frequency between 80 and 2.4kHz, however it was found that some also contain HFC with fundamental frequency between 2 and 12kHz [HO86]. However, these HFCs often have a more directional component to them than the LFCs and because orcas might not be pointing directly at a hydrophone, these are often not

seen in OrcaLab recordings. Another challenging factor in studying orca vocalizations is that two or more different orcas can be making calls at one time, and to determine call identity, these overlapping calls need to be separated [For87].

Several studies found considerable individual variability of orca vocalizations [MB00b] [NSWM06] [PLS04]. A groundbreaking study by Deecke [VDS99] found evidence for cultural transmission of language in orca populations.

In a paper by Miller and Bain [MB00a] they noted that discrete calls always have a LFC, and often end in a "terminal note", a relatively short feature at the end of a call separated by the rest of the call by a rapid change in the slope or frequency of the LFC. Within a call type, they found that the UFC was highly stereotyped. The similarity between call types correlated highly with the how often the matrilinear units (MU) associated with one another, which was from previous work by Bigg et al. [BOE+90].

Recordings often have large amounts of boat noise in them due to the large and increasing amount of marine traffic in the Johnstone straight and around Hanson Island. Many OrcaLab recordings are done remotely using standard marine VHF radios, which makes their quality similar to telephone calls; thus, algorithms that are designed to handle noise are useful [IEE00]. However, these algorithms often depend on harmonic structure, and for many OrcaLab recordings, the orcas are distant from the hydrophones, and there is not much harmonic structure visible on spectrograms.

Deecke describes two different ways to classify vocalizations [VDS99]. The first he calls statistical, and is obtained by looking at properties and statistics of the pitch track. He notes that these methods only assess physical properties of the signal and not how they are perceived. The other method is perceptual, and in this method, humans (or animals) listen to the pitch and classify them. The advantage of this is that factors that are missed by a Digital Signal Processing approach might be detected by the sophisticated auditory systems present in living beings. One downside of this approach is observer bias. Another is that orcas and dolphins have a more advanced peripheral auditory system than humans [WWLA00]. This is of direct relevance to this thesis in that we provide functionality to allow the use of data from both human observers using a perceptual metaphor as well as what Deecke calls statistical methods.

A further contribution from Deecke in the same paper [VDS99], shows results from the Sidewinder algorithm which uses autocovariance of FFT with a neural network. They compare these results to those by humans who had no knowledge of orca vo-

calizations, and show that the algorithm performed well even with poor SNR. The advantages of the Sidewinder algorithm is that it uses more points that other analyses that just use a few points on the pitch contour (min, max, etc. [WWLA00]). The disadvantages to its use are that it had problems when noise had a harmonic component, that it cannot be applied to broadband or pure tone signals, that it does not take amplitude into consideration, and that it is computationally expensive compared to other approaches. The authors suggest that it can be used in combination with other algorithms. This paper showed that using a neural net to classify the output of the Sidewinder algorithm and the perceptual approach using human volunteers gave similar results. Unfortunately, preliminary tests showed that the Sidewinder algorithm is of limited use on the data from the Orchive because of the often limited harmonic structure of the recordings.

In more recent work by Deecke et al. [VAV11], an investigation into the calls of Orcas off the coast of Scotland found that a herring eating group of whales produced pulsed calls like those produced by the NRKW, and that one of their calls appeared to have a herring herding function.

A very early study by Watkins [Wat67] on pulsed calls noted that the highest energy is not always contained in the first, second or third harmonics. This is often seen in the recordings in the Orchive, and this can cause problems with pitch estimation algorithms that assume that the fundamental frequency has the highest energy.

In a paper by Au et al. [AFHA04] the echolocation clicks of orcas were measured and analyzed in detail. They found that the clicks were of very short duration, between 80 and 120us in duration, and had a bimodal broadband distribution, with one peak at 45kHz to 80kHz and one from 35 to 50kHz, 97% of the time. They estimate that orcas would be able to detect salmon at a distance of 100m even in the presence of heavy rain noise. They also mention that boat noise is of a different character than rain and wind noise, and the impact of boat noise on echolocation is unknown. This paper is of relevance to this thesis in that in the data in the Orchive, there are many such recorded echolocation clicks, and a deeper understanding of these clicks and how orcas adapt their echolocation behaviour in the presence of noise could be a useful contribution to the literature. This theme of the impacts of noise on Orca communication can also be found in [HNV$^+$09], where the impacts of noise on the Southern Resident Killer Whale (SRKW) population is examined. In this paper they found that when boat noise increased by 1dB, orcas raised their volume of their calls by 1dB.

Janik [Jan99] studies the use of three different algorithmic techniques to compare dolphin whistles. The first is based on an algorithm by McCowan [McC95] that first normalizes the length of each call, and then takes 20 points along its length as independent variables. These then used as input to a Pearson product-moment correlation matrix which then gives a similarity measure between pairs of whistles, which was then subjected to Principal Component Analysis to reduce the number of colinear variables. These values were then used as input to a k-means clustering algorithm. This paper also used two cross-correlation based methods one that used the cross-correlation of two signals using a method described by Khanna et al. [KGM97], and the other that used the same cross-correlation, but instead took the absolute difference in frequency every 5ms.

Considerable work has gone into the identification of dolphin signature whistles in the last two decades [STWS90]. A recent paper by Janik et al. [JKSW13] presents a new method for comparing dolphin whistles called SIGnature IDentification (SIGID) which uses the observations of human listeners followed by bout analysis, in which they found that they could reliably detect individual dolphins in recordings of groups of dolphins. This research is explained in greater depth of the Ph.D. thesis of King [Kin12] where the use of signature whistles as vocal labels amongst other topics is examined. This is of relevance to the current thesis in that we present an extensible computer system that would allow researchers to conduct the human study portion of the SIGID system using citizen scientists.

Other interesting work on dolphin communication is presented in a recent paper by Janik [Jan13] in which the capability of dolphins to refer to objects and other dolphins using referents and their ability to understand syntactic structure is reviewed. This paper is of relevance in that orcas are of the same taxonomic family as are dolphins, and could possibly share some of their cognitive abilities.

One approach to determining the fundamental frequency of a vocalization is to manually trace spectrogram by hand using appropriate software, for example, in Watwood et al. [WN04] the MATLAB software suite is used to this end. A recent paper by Shapiro et. al [Sha06] uses this same approach. While this approach is appropriate for small datasets, it is most likely untenable for the multi-thousand hour datasets explored in this thesis.

Another approach that has been used is to select peak frequency from sliding power spectrum followed by manual correction to remove octave errors [RL93] [JD94]. In previous work in our lab [NWMT08] this approach was also used by hand with

MATLAB, and in our software suite, we have functionality that allows a user to select a wide variety of parameters for pitch determination algorithms that brings this same functionality to a web-based interface.

One of the most useful algorithms for determining the pitch contours of orca calls is previously described Sidewinder algorithm which uses autocovariance for each spectral slice where the peaks are at multiples of the spacing of frequency bands [VDS99]. This algorithm is particularly appropriate to sounds that are of a pulsed-tone nature. These sounds are composed of a pure tone that is pulsed at a specific, and perhaps varying, rate. A spectrogram of such tones has many bands [Wat67] around a central frequency, with the most energy occurring near the frequency of the tone, and not in the lower harmonics, as is in the sounds of many instruments. The sidewinder algorithm is similar to the spectral autocorrelation method for human speech tracking [LNK87].

A similar algorithm is the Discrete-Logarithmic Pitch Detection Algorithm DLFT-PDA [IEE00]. The DLFT-Makes reliable estimations of pitch and temporal change of pitch from harmonic structure by correlating DLFT spectra from adjacent frames to give reliable estimates fundamental frequency change. It uses Dynamic Programming to create smooth pitch track. However, two drawbacks of this algorithm are that it performs poorly with low SNR and cannot track multiple calls by multiple animals. It was shown to be a a versatile pitch tracking algorithm in recent work by Shapiro et al. [SW09].

Another approach that takes advantage of the banded structure of pulsed calls is an FFT based comb filter method with Dynamic Time Warping [JM06]. In this approach, an upsampled FFT of a sound is produced, and this FFT is then analyzed with a series of comb filters with equally spaced frequency bands to isolate the harmonic structures of different repetition rates. This was explored in [JM06] and then in more detail in [BM07]. This approach takes as input a pitch contour as determined by a pitch determination algorithm, and can take multiple pitch tracks to model sounds in which an LFC is combined with an UFC. The DTW algorithms they explored included the Ellis, Sakoe-Chiba, Itakura and Chai-Vercoe. These vary in the way that the cells are computed, for example in the Chai-Vercoe method, there is explicit handling of insertions and deletions in the cost matrix. These algorithms can be compared to those used in global-sequence alignment algorithms. They found that they could get 90% agreement with perceptual data, and represented a severe test of DTW. Used both the LFC and UFC components.

Another approach used an Artificial Neural Network (ANN) image of spectrogram transformed into an image of data points then normalized and transposed into a vector of points which was the input to the ANN [JHSS93]. Looked at distinguishing calls from each other in one experiment, and whales from each other in another, both experiments gave excellent classification accuracy on a small dataset.

A recent study [Ada06] discusses the use of the Hilbert-Huang Transform (HHT) for analyzing the sounds of marine mammals. They mention that the FFT is limited to the Heisenberg Uncertainty principle, and that time and frequency resolutions must be chosen by the investigator, and relate that the HHT does not suffer from this drawback. They say that FFTs are best suited for analyzing harmonic signals, but not for click sounds, while the HHT is better suited for this. They provide results that show the effectiveness of the HHT over the FFT, especially in the case of clicks where when the ratio between the maximum of the energy when a click is present to that when the click is absent is approximately 20:1 with the FFT and is approximately 100:1 with the HHT.

Another approach using FFTs to classify whale calls can be found in Oswald et al. [OBN03] where they used a multivariate discriminating function analysis for classifying whale vocalizations. In this paper they examine the use of different upper frequency limits (20, 24, 30 and 40kHz) and different variables for each call, including beginning and ending frequency, minimum and maximum frequency, duration, number of inflection points, number of steps and if harmonics were present from the fundamental frequency of each whistle. They got fairly poor classification accuracy (37% at 24kHz) using a multivariate approach with crossvalidation, but their detailed analysis of using these different parameters was quite useful. With a more sophisticated statistical treatment and a dataset larger than the 484 whistles from 29 recording sessions, it might be possible to obtain higher classification accuracy.

Another more recent approach by Deecke [Dee] included the use of the Adaptive Resonance Theory (ART) neural network, a network similar to the Self-Organizing Map approach in that it is an unsupervised approach using a neural-net like foundation; however, in the ART algorithm, if a new pattern is found that is sufficiently different from existing patterns, it becomes the reference pattern for a new category.

Another way to classify audio features is to use the Pearson product moment correlation which was used in Watwood et al. [LLS04] to classify the vocalizations of bottlenose dolphins. The main advantage of this technique are that it takes into account the differences that are lost when whistle loops are normalized in time, it

also ignores absolute frequency and bandwidth information. By doing so, it makes loops that have monotonic changes in frequency appear more similar even if they are considerably different in absolute frequency.

2.2 Systems for working with large audio corpora

A different methodology of doing audio feature extraction was explored in the paper "Distributed audio feature extraction for music" [BT05]. In this paper the authors use a dataflow architecture for distributed audio feature extraction on a set of networked computers. A dataflow architecture allows users to connect small processing algorithms in a graph like network and to then use this constructed network to transform input data into a desired type of output data. A simple example of a Marsyas network would take audio samples from a file on disk, multiply them by a number and output them to another file on disk, which would have the effect of increasing the volume (loudness) of the file. However, most of the commonly used algorithms in Marsyas are complex feature extraction and machine learning algorithms. Marsyas was an existing software platform that used this dataflow architecture, and this paper explored the use of dataflow architectures on networked computers. One of the important contributions of this paper was that the optimal vector size of data to send from one computer to the other was influenced by the network settings of their Ethernet routers, but in their case a 256 size vector provided the best speed. Results are also presented to show that by using a Collection Partitioning Adaptive algorithm the authors were able to get the best utilization of their cluster. This algorithm splits up the files to be processed and adaptively sends more jobs to the computers that have completed their past jobs quickest. The authors also note that it is very simple to port an existing Marsyas network to this distributed framework, and there is indeed current efforts ongoing to do this in an automated fashion.

One serious drawback of this framework is the large amount of disk and network traffic that is generated by the producer nodes on these system, the authors point out that the maximum number of worker computers that could be served by one dispatcher computer was only four. This makes sense because of the large amount of data generated in many of the intermediate steps in the audio feature extraction pipeline. The authors propose a system of hierarchical dispatchers and workers to solve this problem.

For many interesting tasks in realtime computer generated music and analysis of

music in performance situations this type of system would be very useful, for example, one dispatcher could communicate with a number of workers each of run a different type of Machine Learning algorithm and where the results are then collated. However, for the present task of analyzing huge collections of audio that is already on disk in a batch fashion, the architecture in distributed Marsyas likely would need further work to be of much use.

In "Overview of OMEN"[MMF06], McEnnis, McKay and Fujinaga describe their system called OMEN (On-demand Metadata Extraction Network), a system that shares many commonalities with the Orchive system we describe in this thesis. The main issue their system aims to overcome is copyright problems that researchers encounter when they want to legally extract audio features from songs. In the strictest legal sense, each researcher must purchase their own version of a song in order to do any kind of audio science on it. This becomes prohibitive in the age of the Million Song Dataset. In the OMEN system, a coalition of libraries creates a network of systems that host the raw audio data of the song, but only send back to the researchers the specific audio features that they are interested in. Another interesting aspect of this system is that they propose to use the unused computing cycles of library browsing computers to be a grid computing resource for doing their audio feature extraction.

One big difference is that their system is primarily based on Java, where the system described in this thesis is primarily based on Python, C++ and Javascript. In many ways, the Java ecosystem is enticing for building server applications, and several times in this project we considered doing feature extraction in jMIR, the Java Music Information Retrieval library that the OMEN system happens to use. However, there are considerable barriers to getting a productive Java development and production system installed and maintained.

2.3 Intelligence Augmentation

This archive of data is now available in electronic form which makes it easier to access than when it was on a single set of analogue tapes at OrcaLab, what would make it even more useful to scientists would be the ability to collaborate together on the process of annotating audio, running experiments and analyzing results. This type of work falls under the rubric of Computer Supported Collaborative Work [BS91] (CSCW), a field which studies multiple individuals working together with computer systems.

In this particular case, we have a number of different communities who have shown interest in this archive, these include Orcalab and its collaborators, whale biologists, developers of bioacoustic and Music Information Retrieval algorithms, and the part of the general public who enjoy listening to whales and might like to help the scientists who are working on this project.

As the scientists in this project are located all over the world, having a system that enables them to work together collaboratively even though they are separated by large distances will be a worthwhile challenge [BM02] [OO00]. There are a number of pitfalls that have been identified [Gru88] including the disparity between who performs the work and who gets the benefits, the breakdown of intuitive decision-making and the difficulty in evaluating the application. To overcome these problems, we are eliciting feedback at every stage of the design process from stakeholders, including the members of Orcalab, and we incorporate this feedback using an iterative development methodology. However, even with these efforts, it should be noted that the community that receives the benefits, which is primarily computer scientists along potentially with population biologists and linguists, is different from the community that does the work, which is the biologists who know the vocalizations of the species of interest. The recruitment of biologists with knowledge of orca vocalizations has been one of the most challenging parts of this project.

Our system uses two types of web based interfaces. The first are tools aimed at expert users, and the second are simpler interfaces designed for crowdsourcing the annotation. There are a number of tools that experts use to segment and analyze audio and specifically bioacoustic data. One of the most popular is Raven (http://www.birds.cornell.edu/raven), a toolkit developed at the Cornell Lab of Ornithology. Our expert based tools have many similarities to Raven such as the ability to view waveforms and spectrograms at multiple levels of detail. In addition, the system described in this thesis tightly integrates the visualization and viewing of data from machine learning classifiers.

Another tool is the Sonic Visualizer [CLS10], which supports a wide variety of waveform and spectral audio representations. The popular audio program Audacity has been extended to allow for the annotation of audio data [LBF06]. The biggest difference our system compared to these systems is that the software is all web-based, so users do not have to install a separate program and can more easily view long audio files and analyze data across multiple recordings.

The goal of early research in Artificial Intelligence (AI) was to build indepen-

dently intelligent agents that could perform cognitive tasks. There were a number of approaches to build such systems. Some researchers, such as John McCarthy, chose to build symbolic manipulation systems where knowledge was represented by formal logic. Others, such as efforts by Marvin Minsky, instead attempted to combine several ad-hoc solutions to create an independent agent. Most of these systems failed to take into account that in the real world. Intelligent agents, such as people, are embodied, that is they have a physical form, and this physical body interacts both with other agents in social interactions as well as with its environment.

In the 1980's instead of Artificial Intelligence, researchers instead began to investigate Intelligence Augmentation [FN92], where instead of trying to build intelligent systems that were designed to replace humans, researchers started building systems that would help people to solve problems, combining the best of human and artificial problem solving strategies. These agents were designed to assist humans and inherently carried within them the idea that intelligence was embodied and interacts with people and other artificial agents through social interactions.

A interesting study was carried out by Sumner [SBHK97] and investigated embedding agents that functioned as critics into a design environment. In a critiquing approach, an artificial system communicates in a dialog with a human, suggesting changes to places where there are problems with the design. The system described in this paper is that of voice menus for a phone-based interface. The system is designed by a person or team, and the system critiques the design in places where the designer has violated previously determined rule-based constraints. This work was extended in a subsequent article [FNO$^+$98] where a system to help designers create kitchens was described. One interesting result from these studies was that designers reacted negatively to being critiqued, and began to anticipate the response of the system and took steps to avoid being critiqued. This highlights the importance of making sure the system that is built makes the users feel that they are empowered and are being supported by the system.

In a debate entitle "Direct manipulation vs. interface agents" [SM97], Ben Schneiderman and Pattie Maes debate the relative advantages of direct manipulation and software agents. Direct manipulation is described to be user interface techniques where large amounts of data are presented to the user on the screen at one time. Through a user interface with fast feedback (¡100ms), the user navigates through the data. Software agents, on the other hand, are described to be long-lived software systems that learn user preferences and adapt to users, presenting them with data

that the system estimates will be useful to the user. In this paper, these two approaches are discussed and contrasted. It is interesting to note that by the end of the debate, both sides acknowledge benefits to the others position, and indeed, as we have seen in the decade since this debate, systems are being designed that support both direct manipulation and software agents. In The Orchive, we are attempting to design a system that supports both of these paradigms, allowing the user to directly manipulate large quantities of data, and also provide software agents that can mediate interactions between users, other users, annotations from other software agents, and the data.

In the paper "Distributed intelligence: extending the power of the unaided, individual human mind", [Fis06] Fischer provides an excellent overview of the field of distributed intelligence, relating it back to work in distributed cognition by Hollan et. al [HHK00] as well as tying in other concepts, such as those of Intelligence Augmentation, advanced visualization interfaces that exploit the power of the human visual system, making user specific systems, and exploring computing off the desktop, which includes both large scale and small scale displays. This paper also explores the difference between "Tools for Learning", which have the goal of educating the user so that they will be independent of the system and "Tools for Living" that help people do things they could not do themselves. In The Orchive, we are developing tools to support both these paradigms. First, to help users learn the calls used by the orcas, and second, to help more advanced users interact with the data, allowing them to ask and answer research questions.

In the paper "Collaborative, programmable intelligent agents" [NMW98], a system that extracts semantic data from documents using intelligent agents is described. This paper talks about the difficulty of creating systems that approximate real intelligence and instead proposes that that we should investigate what tasks computers are good for, what tasks people are good for, and how they can best complement each other. Their goal is to create a system that works like a reference librarian, taking imprecise requests from clients and returning relevant results. They then present a system called "Apple Data Detectors", which uses a collaborative system to work with the user to fulfill their requests.

The field of Distributed Cognition studies the social and environmental aspects of cognition and realizes that thinking is not just something that happens in isolated minds, but can also take place in the interactions between minds and their environment. It was first formulated into a discrete discipline by Edwin Hutchins, which is

described extensively in his book "Cognition in the Wild" [Hut96] after his research into the native people of Papua New Guinea and their system of public litigation [Hut80]. In a classic paper by Hollan, Hutchins and Kirsh [HHK00] distributed cognition in different environments is described. One of these is on the bridge of a ship, a location where multiple persons have to work together to accomplish many simultaneous and shared goals. In this situation, people work together, sharing information via intentional and consequential communication and also offload cognitive tasks to their environment by the use of maps, displays and other physical objects.

The practice of Cognitive Ethnography [HHK00] is often used to study distributed cognition. It studies the cognitive processes that underlie the work that occurs in an environment, and takes into account the effect of the social context of the participants as well as the way the physical world affects the process. Ethnography is a methodology in which researchers study a system by making direct observations in as natural a setting as possible [McG95].

In the book "Distributed cognitions: Psychological and educational considerations", [Sal97] a number of authors investigate distributed cognition, and give examples of this in a number of fields, including daily life and education. One central theme of this book is that people think in conjunction or partnership with others and use tools to help them think. They also note that it has been observed that the performance of people in teams supported by computational tools is often superior to people working alone. They then investigate how distributed cognition can be harnessed in the field of education, with the goal of helping students to learn more effectively.

There has been a long and fruitful history of the application of distributed cognition in the sciences. If one adopts a loose definition of distributed cognition as thought processes that happen not just in one individual brain, but are instead distributed through a community and are mediated by artifacts, the whole history of science itself, even back as far as Aristotle, has been an example of distributed cognition. In the words of Isaac Newton:

"If I have seen further, it is by standing on the shoulder of giants".

However, recent advances in the understanding of the concept of distributed cognition from researchers in the social sciences, combined with advances in computer and communication technology could dramatically increase the speed, facility and ease of communication between scientists. These increases could potentially be revolutionary changes in certain very difficult problem domains. In such domains, the

problem itself is too big to be solved by one single researcher and must instead be solved by teams of researchers with the aid of artifacts that aid cognition.

There are many different such problem domains that require collaborations between large numbers of users, three such examples are high energy physics, genomics research and the study of the vocalizations of whales.

2.4 Citizen Science

Citizen science is a process by which members of the public are involved in a collaboration with professional scientists to collect and analyze data. Although its coinage as a term is rather recent, it has a well-established history, with one of the earliest examples being that of Linnaeus who in his Systema Naturae [Lin58] asked volunteers from the public to send him botanical samples from around the world for the development of his taxonomical system. Other examples include the establishment of a system of amateur weather observatories across the US by Thomas Jefferson [Fie09] and the Christmas Bird Count [LCN+09] of the Audubon society and Cornell Lab of Ornithology.

However, in recent years the use of networked computers has dramatically accelerated the ability for scientists and the public to collaborate on large scale projects. The term crowdsourcing refers to a new type of collaboration where non-specialists help expert scientists [How08] and has been used to great advantage [Sur05] in a number [Bra08] of research programs [Tra08]. One of the most successful examples is Galaxy Zoo [SLB+08]. Games-with-a-purpose (GWAP) [vAD08], which are computer games that harness the ability of people to solve tasks in a game setting. The first GWAP was the ESP game [vAD04] in which two users on computers connected to the internet try to guess words that describe an image. In "Game-powered Machine Learning" [BTL12], Barrington, Turnbull and Lanckiet describe a system that combines Games With A Purpose (GWAP) with Machine Learning in a framework that they call "Active Machine Learning". In the paragraphs below, we examine these contributions in greater detail.

One of the most successful citizen science programs is Galaxy Zoo [SLB+08]. In this project, astronomers had collected images of many thousands of galaxies and wanted to characterize them by the chiral handedness of their spiral structure, that is, were they spinning clockwise or counterclockwise? It was assumed that there would be an even distribution of these the two chiral hands, left and right, and any

deviations from this would be an important and surprising result. Even the most advanced current computer algorithms are not able to categorize galaxies based on their handedness, but humans can do this classification easily. In this paper, the authors describe their system and results, and present results that indicate that there is a hint of positive correlation for galaxies nearer than 0.5 Megaparsecs.

Another research program that benefited from crowdsourcing was the Stardust@home project [BM06]. Stardust [ABD+97] was a NASA space mission that flew a spacecraft through the tail of comet Wild 2, collected dust from the comet and from interstellar matter using an aerogel, and then returned the satellite to earth. Comet dust was collected on one side of the aerogel, and the other side of the aerogel was exposed to interstellar matter during the entire mission. The analysis of the particles from the comet were straightforward due to the high number of particles, but the analysis of interstellar grains was much more difficult due to the small size and number of particles. The Stardust@home project allowed users from around the world to signup on a website and interactively view and annotate microscopic images of the aerogel. There was overwhelming participation by the public, and they were able to generate results that were useful to the scientists on the project [ABD+97].

There have been a number of articles that investigate the benefits of crowdsourcing. Hong [HP04] presents results that show that a group of problem solvers with a diverse background can outperform smaller groups of experts. This is of interest in the current work because there are only a very small number of orca vocalization experts in the world, but there is a large number of people who are interested in listening to orca calls, as is evidenced by the traffic on the Orca Live `http://orcalive.net` forums.

A recent article [KCS08] describes crowdsourcing with the Amazon Mechanical Turk system, a web based system where people can sign up to work on small tasks in return for micropayments. The advantage with using the Mechanical Turk system is that because people are paid for their work and have to pass a scientist defined test, the results obtained might be of higher quality. The drawback to this system is that the workers must be paid. Most scientific projects have limited budgets, and therefore this type of solution is often not practical.

One exciting new development in crowdsourcing is games-with-a-purpose (GWAP) [vAD08], which are computer games that harness the ability of people to solve tasks in a game setting. The first GWAP was the ESP game [vAD04] in which two users on computers connected to the internet try to guess words that describe an image. If

both users guess the same word, this word is then added to the tags to that image, and is also added to a list of forbidden words which forces people to choose new words to describe an image. This game has proven immensely popular, with many millions of tags added to images. Another such game is Tag-A-Tune [LvA09], an input agreement game that has people add tags to music clips. These games, and others like them, exploit the powerful reward system of the human limbic system and are an excellent way to hold the attention of large numbers of users and generate high quality data. The challenge with games-with-a-purpose is to develop a captivating game experience, but with time and skill, they perhaps hold the largest potential in terms of sheer size of population for performing crowdsourcing.

Although many of these projects concentrate on recruiting large numbers of participants, even small numbers of human volunteers can be of great use when analyzing bioacoustic data. Of particular relevance to this thesis is work done by researchers in cetacean bioacoustics [For91] [YFM+10] [Yur05] [Kin12] where human volunteers were used to classify orca and dolphin vocalizations into distinct categories. These researchers used different kinds of technology to do this, primarily not with a computer, but in recent work by Yurk [YFM+10] [Yur05] a web-based interface was developed to allow volunteers to help classify vocalizations. The results by the human volunteers was used to help evaluate the performance of machine learning algorithms. This has direct relevance to the current work in that in this thesis we describe a system that extends this work by providing an integrated system where researchers can upload, annotate and create new sets of classification tasks for humans to perform. The system that we have implemented also adds game like elements to hopefully provide greater engagement of the public along with a concomitant increase in the amount of data collected.

In "Game-powered Machine Learning" [BTL12], Barrington, Turnbull and Lanckiet describe a system that combines Games With A Purpose (GWAP) with Machine Learning in a framework that they call "Active Machine Learning". In this framework, the user is presented with a set of plausible answers by a Machine Learning system, these plausible answers are chosen to be as close to the actual answer as possible, which helps the system train itself on difficult data most effectively, and also makes the game more fun and challenging to play, since the answers are so close to each other. This kind of system directly inspired the work we describe earlier with GWAP, and how we are developing a system to let users help us to classify orca calls, and by using active learning we can help to make this game fun and challenging.

Chapter 3

Software and Systems

The data in the Orchive presents challenges to the tools and workflow typically used in bioacoustics and Music Information Retrieval. Each of the recordings is 45 minutes in length and at a sampling rate of 44100 samples/sec means each file is 484MB in size. Tools such as Audacity, Sonic Visualiser and Raven take a considerable amount of time to load and are noticably sluggish to navigate inside.

However, the larger problem is the sheer number of recordings, the Orchive currently holds around 20,000 of these 45 minute recordings, and is continuously growing over time. These recordings take 17TB of disk space, and even now in the era of 4TB hard drives, distributing this data to each scientist interested in studying this data would be difficult.

Even if this data could be sent to each scientist, the classifications and annotations created by each researcher somehow need to be shared if they are to collaborate effectively. Existing tools do not facilitate this process, and require researchers to use ad hoc methods, such as emailing text annotation files to each other.

There are large numbers of people in the general public that enjoy listening to orca vocalizations, for example, the Orca-Live website (http://orca-live.net) streams audio directly from the hydrophones at OrcaLab, and even with the relatively rare occurences of orca vocalizations, they can have up to 20 people listening to the stream at once around the world. Enlisting these listeners to become citizen scientists and to help annotate the large amount of audio from the Orchive could be a rich source of human annotations.

Another large problem is that extracting audio features and doing machine learning on this amount of data would be impractical on a single computer and requires the use of large numbers of computers connected to high speed storage to process in

a reasonable amount of time.

In order to address these challenges, we have developed a system called The Orchive where expert scientists can listen to, view and annotate audio, view the output of audio feature extraction algorithms, run and view the results of machine learning systems on this audio, and to enable citizen scientists to provide annotations that can be used by both the machine learning and scientists.

In the following section we first briefly describe The Orchive version 1.0, a first attempt at developing a system to provide these functions. We then discuss the issues we discovered after 5 years of scientists using this system, and then describe version 2.0 of the Orchive, a completely rewritten system which takes into account the deficiencies of version 1.0 and tries to address them.

3.1 Choice of Platform

In 2007, a project was begun to develop a system that would allow users to view, listen to, and annotate the data in The Orchive. At that time, there was only 3TB of data, and most of the recordings had not yet been digitized.

The first decision that needed to be made when developing tools for the Orchive was the choice of platform and toolchain to develop it with. We had a number of requirements for this system. Firstly that it allowed researchers to access all the recordings in the Orchive. The second was that it allowed them to annotate the recordings to show where regions where orca were vocalizing, and to further annotate these vocalizations with the type of vocalization being made, be it echolocation, whistles, pulsed calls, voice notes, boat noise, and the many other sounds in these recordings, including marine mammals such as seals, dolphins and humpback whales. Labeling the pulsed calls with the call types from the John Ford catalog and even with the pod, matriline and individual identity of the whales were important. The system should also allow scientists to view the wide variety of other data collected by OrcaLab, including the 30,000 lab book pages, maps of orca travel, incidence reports, comments about recordings, call catalog, and the time and date of each recording. Finally, this system should allow researchers to run audio feature extraction and machine learning tasks and to view these results within the interface.

One requirement that was added later was the ability to allow citizen scientists to help annotate data using a simple game-like interface. The annotations from the users should be accessible to machine learning algorithms as a way to provide more

training data.

Although it would conceivably be possible to develop a standalone program running on individual machines that would meet all these requirements, the difficulty of sending the audio to the users, collecting and merging their annotations, and running large-scale audio feature extraction and machine learning algorithms made this approach impractical.

A platform that would allow all these requirements to be met would be a web based platform. In this paradigm, all of the data resides on a central server, and a web server provides a link between this data and users who connect using standard web browsers. By storing all the data in a central location, one large server with an array of disks can store and backup the large amount of audio files. The web server stores the annotations from each user and allows other users to view and use these annotations, and to view the results of audio feature extraction and machine learning jobs.

3.2 Orchive Version 1.0

When this project was conceived, the set of toolchains for developing rich web interfaces was considerably less advanced than it is now. Most web applications consisted of small amounts of Javascript on the client which communicated with the server using Asynchronous Javascript and XML (AJAX). One of the most popular server platform for developing web applications was Ruby On Rails (RoR), a framework that allowed for good modeling of problem domains using an Object Relational Mapping (ORM) system combined with interfaces to allow the server and client to interoperate using AJAX using the language Ruby.

One of the primary web applications that enabled people to see large amounts of data was Google Maps. In Google Maps, all the levels of zoom that are available are precalculated and stored on servers, which allows for rapid transmission of these resources to users. This paradigm of precalculating data allowed for higher speed and more interactive website than one that would build each map on the fly.

One large hurdle that existed at the time this project was developed was the lack of support for audio on web clients using pure HTML. The only solution that existed for developing rich audio applications on the web was Flash, a technology developed by Adobe that allows developers to develop a program that could be embedded in a web page.

Figure 3.1: A small annotated section of audio from the Orchive

For the Orchive version 1, we developed a system where data was stored in a MySQL database, and was accessed using the ORM layer in Ruby on Rails, which built web pages and sent these to clients. The web page was primarily HTML, but contained a rich web application written in the HaXe language, an open source platform that could build a Flash application as one of its possible targets. The audio was stored on the server, and in order to provide the fastest possible interaction, spectrograms and waveform displays of this audio were precalculated using Marsyas and were stored on the server at a number of different zoom levels. The Flash application allowed the user to listen to, view and annotate the recordings.

This system also allowed for the user to run different audio feature extraction and machine learning algorithms, and to view this data and classifications overlayed on the spectrogram. Following best practices of the time, this data was also stored in the MySQL database. An image of the website is shown in Figure 3.2 which shows a section of one recording with annotations.

This interface is still in use, and since July 2008, there have been approximately 3,900 people who have viewed the site with over 10,000 individual visits. There are 42

scientists registered as users on the site, and of these, 11 scientists have contributed annotations. There are currently over 17,000 annotations in the database, most of which label if the clip is an orca vocalization or background noise, but with also with almost 4000 that contain which type of orca call is being vocalized. Some annotations even contain the pod and/or matrilinal identity of the call.

A version of this web interface was also developed to view the data from the VENUS and NEPTUNE projects. Dr. Richard Dewey, an Associate Director of VENUS was able to use this web interface to find the first examples of whale vocalizations in the VENUS database. The existing VENUS interface allowed users to only see one small 5 minute chunk of data at a time, and because the spectrograms were not precalculated, it took considerable time for each new 5 minute section of audio to load. Our system allowed him to quickly scan over large numbers of recordings to find whale vocalizations.

Even though the interface was in some ways successful, over the years many drawbacks to this system were identified as people used the interface and new features were added to the site.

The primary deficiency was that this system required all the spectrograms and waveform images to be precalculated. Although this allowed for a substantially more responsive interface, as new recordings were added to the database, the entire precalculation process needed to be repeated. The last rebuild of these images took over 3 months of continuous processing time on the server. Another downside is the large amount of disk space required to store the spectrogram images, at the current time, these images take up over 1TB of disk space for 17TB of audio data.

Another issue with this system is that over time, it became apparent that there were a number of problematics issues with the software tools used to create the system. Even though RoR is still widely used, there are a number of issues with the performance of the Ruby language, especially when doing other tasks than simply serving web pages. In addition, because of its relative obscurity, there are issues with APIs for interfacing Ruby to other tools. It was found that the Python language has better interfaces to other tools, such as Marsyas, so much of the new code that was written was written in Python. This allowed for easier development of new features, but increased the number of languages required to be used, which complicated development time.

In addition, over time the popularity of the Flash platform declined dramatically, both in terms of acceptance and use by the general public as well as within the

web programming community. The fact that Flash could not be used on Apples iOS devices was one of the contributing factors, but the larger factor was the development of standards based web interfaces that are not controlled by a single company, but are formed from a consensus among members of the community. Because of this, new development on the site was done in Javascript, for example, the call catalog was developed as a Javascript application. Interaction between Javascript applications on the page was straightforward, but interactions between the main "orcaannotator" Flash application and Javascript were often challenging, and became more so as the complexity of the application grew.

For many tasks in the Orchive, it was found that neither Python or Ruby was fast enough, and certain aspects of the site were written in C++. For example, the programs to do the calculation of spectrograms was written in C++.

In the end, it became very difficult to add new features to the website because of the need for development to occur simultaneously in Ruby, Python, Flash (HaXe), Javascript and C++. At the current time, the Orchive version 1 website contains 11,296 lines of Ruby, 3949 lines of HaXe, 987 lines of Javascript, 3405 lines of Python and 2520 lines of C++. The number of lines of code was manageable, but the number of languages required to add new features became difficult to manage.

One of the most dramatic issues with this was demonstrated in 2010 when developing the VENUS version of the website. In order to launch these websites, a number of new features were required to be added to the site to fulfill requirements of the VENUS team. Some of these were straightforward to add, but there were a number of features that were very difficult to add given the complexity of the project, and we were not able to add them in a reasonable amount of time and were, therefore, unable to launch the website to the public at large. In addition, version 1 of the Orchive did not contain the functionality to separate different datasets from each other, so for both the VENUS and NEPTUNE versions of the site, all the code had to be cloned and three separate versions of the code had to be maintained and when bugs were fixed in one version, this same fix had to be applied to all codebases.

One big issue that was identified later in the project was that many of the scientists who would like to participate in the project were in remote locations with limited or no internet access. For example, some of the scientists had limited time to work on annotating data when in their offices, and had more time when they were on board boats, studying orcas. With the traditional AJAX development paradigm of having a thin client and most of the code being on the server, the assumption is that the

client has continuous access to a high-speed internet connection.

Another issue with the website is that in order to support a wide variety of annotations, it was decided early on to have annotations be in a free form text field. The hope was that scientists would develop their own consistent vocabulary of annotations, and that an open system for annotations would allow for both a standard vocabulary, for example, the calls from John Fords call catalog, as well as being flexible enough to accommodate new annotations. In a number of ways this was successful, many of the annotations are in a consistent format, and there are a number of interesting annotations that would have not been supported by a fixed vocabulary like "amazing sequence" and "great N4". However, the lack of a consistent vocabulary became a major hurdle when trying to convert these annotations into a form usable by a machine learning system. For example, there are 383 "N1" annotations, 59 "N01" annotations and 20 annotations that contain either "N1" or "N01" within them, sometimes preceded by a space or appended with the matriline or a note about the call. Writing scripts to convert these into a consistent vocabulary became progressively more difficult.

For these reasons, amongst others, it was determined that a new version of the website should be built using more modern technology and using a smaller number of programming languages and tools.

3.3 Orchive Version 2.0

In 2012, a new version of the Orchive software was begun from scratch. For this new version, it was determined that the Python programming language would be the main scripting language used, that Javascript and HTML5 would be used for all client facing software, and that C++ should be used where needed for doing processing of large datasets on the server. This allowed us to concentrate development in Python and Javascript, which greatly simplified development.

There were a large number of web server platforms available that were written in Python, and after extensive research and testing, it was determined that Django provided a good blend between power, popularity and ease of development.

For the front-end, the existence of new tools and frameworks for developing rich Javascript made it feasible to write larger and richer web applications. One of these was the Backbone framework, a toolkit that allowed for the development of MVC (Model/View/Controller) applications in Javascript on the client, which allowed for

the creation of applications that could work well both in environments with a high speed internet connection as well as situations with only periodic access to the internet. One of the exciting things about this kind of framework is that it allowed for the development of offline applications that could be deployed as tablet or phone based applications using PhoneGap, a framework that allows developers to build iOS or Android apps using HTML5 and Javascript.

In addition the standardization and wide deployment of HTML5 audio in both the Audio DOM (Document Element Model) element as well as the Web Audio API made it possible to develop applications that could reliably play back sound, a vital part of our application.

One of the big issues with the first version of the Orchive software was the choice to allow for free-form input of annotations by scientists. In the new system, we changed this to only allow annotations from a fixed vocabulary. However, the system supports an arbitrary number of such vocabularies, and researchers can add new entries to these dictionaries. To support the previously allowed free form annotations, we add a comment field to the annotation which can contain whatever text the researcher desires.

Another issue that we came across with the first version of the website was the need for all the spectrograms, waveforms and other assets to be pre-generated. In order to overcome this, we took advantage of the Marsyas bindings to Python, and wrote code to allow the web server to generate spectrograms from audio on the fly. In order to improve latency, the Javascript interface sends multiple requests for rectangular sections of the visible spectrogram, which allows multiple cores on the server to generate spectrograms in parallel. In addition, these spectrograms are cached in memory on the server (using memcached) to speed subsequent requests for the same region of a spectrogram.

Although not all the features of the old site are yet supported in the new site, the majority of them are, and the new site contains functionality to display audio features not present in the old site. There are 2849 lines of Python code in the new interface, and 4230 lines of Javascript, which shows the dramatic move from a server based application to a client based application.

Although both versions of the website use a REST (Representational State Transfer) based API, the new Javascript code uses this REST API exclusively for communication of state between the server and client.

In order to deal with the issue that the code had to be cloned for different projects,

the new version of the website allows for multiple projects to co-exist within one set of web server code. Currently the system contains three different projects, the Orchive, a site with bird bioacoustic data from the Alberta Biodiversity Monitoring Institute and a project dealing with partially annotated chant data from the Torah and Koran traditions. Because of this, bug fixes and updates made to one project are reflected in all the other projects. The web server displays different headers and graphics for the different projects and supports different domain names URLs to access the data which allows the different projects to maintain their identity.

This new server software also incorporates the citizen science games directly in the website, and allows for the easy development of new game levels directly from the annotation data.

3.4 System Overview

Given the vastness of this archive, there are certainly many different tasks that bioacousticians and other scientists might want to do with it, but one primary obstacle is that the Orchive is currently mostly unlabeled. After 4 years, there are approximately 12,000 annotations in the Orchive, which represents only a few hours of actual annotated time. Even with many scientists working on Labeling this data, it would take many person-years to label the entire Orchive. By using a combination of expert users and Machine Learning software, it is possible to train Machine Learning classifiers on expert labeled data and from there to predict the location of orca vocalizations. In order to further add to the training data for the Machine Learning classifiers, we have also built simple iPad and web based casual games with a serious intent, which we call serious casual games. Casual games are those like the well known Solitaire, which take a short amount of time to play and do not require all the users attention. By allowing these citizen scientists to assist us, we are able to collect larger amounts of data, and to sort through data we have generated quickly. In fact, one of the main users of this game software has been ourselves, we find the game interface a quick and effective way to label training data.

With these interfaces, we are able to label substantial amounts of data, but are still far away from Labeling the entire 20,000 hour archive. In order to do this, researchers have commonly used a combination of the techniques of Audio Feature Extraction and Machine Learning to classify audio. In this thesis, we will use a combination of Audio Feature Extraction and Machine Learning to identify sections of the Orchive

Figure 3.2: System Diagram

where there are clear orca vocalizations that are distinct and not overlapping with the vocalizations of other orcas and are relatively free of boat noise. These extracted vocalizations will then be made available to the scientific community and will also be used by our lab in future work for answering other scientific questions about this species. A system diagram of how this process of Machine Learning, Expert Interfaces and Serious Casual games works, please refer to Figure 3.2.

In order to do this, we will compare a number of different solutions for doing distributed audio feature extraction and Machine Learning and will compare not only their performance in classifying audio, but also our experience with using these tools.

For my evaluation, we will generate two different types of results. The first type will be the output of the audio feature extraction and Machine Learning task. For this task, we will look at the distribution of orca vocalizations in the dataset, and will train Machine Learning classifiers that can distinguish regions of the Orchive that have individual orca vocalizations.

In order to analyze the audio in large online archives efficiently, a scenario involving distributed computing must be used. There are many different types of systems that perform Machine Learning and many of these can distribute their work to different machines. Some of these systems are better suited for this task than others, and in this thesis we investigate a few combinations of these tools.

One of the possible solutions to the problem of the segmentation of audio into orca vocalizations and other sounds is to use a Machine Learning supervised classifier

system. These systems require hand labeled training data, in our particular case, audio would be labeled as either orca or background. This training data is then used to train a classifier system that can classify future instances based on features of the training data. There are a wide variety of classification algorithms that are commonly used, some of the most popular are Support Vector Machines (SVMs), Neural Nets, Decision Trees and Logistic Regression. In our past work we have found good success in using SVMs, and have shown that SVMs typically outperform many other Machine Learning algorithms. However, there is current work in this area showing the benefits of many alternate algorithms for different problem domains.

There are two steps to the problem of the classification of audio signals. The first is that of audio feature extraction, and the second is that of classification with Machine Learning systems. Sound is a pressure wave in the air, and the variations in pressure over time can be stored in a computer as real number values. This raw waveform data contains all the information of the signal, but because of its extremely high dimensionality, is difficult to analyze with machine learning systems. To take this raw data and to put it in a form more conducive for analysis, we generate a large number of higher level features, including spectral features [Tza08], pitch information [dCK02], chromatic scale information [Tza08] and other features based on models of the auditory cortex [Lyo82]. One can also model the statistical behaviour of these properties as they change over time, an avenue that has been shown to be fruitful [Tza08]. In our system we investigated the use of a number of audio features, the results of which are discussed in the Evaluation section, but in summary we found that using all the standard audio features was most useful, however, using pitch information degraded performance of the classifier. This is a surprising result and one that we are investigating further.

The second step is to take these audio features, train and test a Machine Learning classifier on them, and then run this Machine Learning classifier over all the data. In many cases it is beneficial to pre-calculate the audio features for the data. Once good parameters can be determined for the audio feature extraction engine, such as the window size, hop size for the spectrum calculation and the memory size for the windowed statistical calculations, these features can be saved and reused in multiple Machine Learning experiments with different parameters. One example of this is the recent release of the Echonest audio features for the Million Song Dataset [BMEWL11]. This collection of audio features for one million popular songs took a considerable amount of time to process, and one approach is to save these audio

features and running different machine learning algorithms or different parameter sets of these algorithms on them. Another point that should be made here is that the form of the features that are extracted in this step can have a direct relation to the performance of the subsequent Machine Learning step. Matching appropriate audio features with suitable algorithms often requires an extensive search process and is facilitated by experience on the part of the individual researcher.

It should be noted that both of these steps, the audio feature extraction and the Machine Learning could be carried out in the same executable. There are a number of advantages to this as well as some disadvantages. One major advantage is that the intermediate audio feature data does not have to be saved on disk. This intermediate data can be very large in size, and can be many times larger in size than the original audio data, depending on the type of compression used for the original audio. In many cases in the real world, it is less expensive to just recalculate the audio features each time that a new Machine Learning run is carried out rather than saving them on disk. This also can help to overcome any versioning problems, where different versions of audio features are combined together in the training, testing or predicting stages. It will be a balance between these two poles for production systems to manage.

For this work, Marsyas was used for doing Audio Feature Extraction. There are other Music Information Retrieval toolkits for extracting audio features, but the audio features output by Marsyas are typically amongst the most robust and high performing in the literature [Tza08]. We have implemented this feature extraction using native filesystem operations loading data off of NFS or a local disk depending on the cluster situation.

In this thesis, the Westgrid collection of computational clusters is used for running the huge audio feature extraction task. This uses a very simple queuing system known as Torque, which is the latest evolution of the ancient PBS (Parallel Batch System) parallel job distribution system. PBS allows scientific users to schedule large jobs to run on computers, and performs well for certain scientific tasks where a single, usually large, program is run on a variety of different datasets, and the results are then saved to disk and later analyzed by a scientist. This type of computing was traditionally known as Grid Computing. While this system has certain benefits for running specific scientific tasks, for other tasks, the methodology of having separate computers running mostly independently, and all coordination of tasks being the responsibility of the programmer with an tool such as MPI quickly becomes difficult. The MapReduce paradigm helps in the coordination of large parallel tasks, and the

Hadoop system allows programmers to efficiently write distributed programs that run on huge numbers of computers and allows for failures of individual computers.

The primary task I will investigate in this project will be identifying regions of the Orchive with distinct orca vocalizations that are nearby to hydrophones and isolated from other orca vocalizations.

3.5 Machine Learning

In addition to the presentation and collaborative annotation that the Orchive supports, we have also developed a set of Music Information Retrieval and Machine Learning tools that are available for researchers to run on the data in the Orchive. All of these tools are part of the Marsyas [Tza08] MIR framework

The first set of tools are audio feature extraction tools that have been adapted from the field of Music Information Retrieval (MIR) [FD02] to the study of orca vocalizations. In particular, we have added support for the generation of spectral statistics including FFT spectrum, MFCC coefficients, Spectral Centroid, Flux and Rolloff.

We also support classification of audio using Support Vector Machines [ME05], an advanced type of Machine Learning technique that finds optimal hyperplanes in high dimensional datasets, which we then use to classify audio. This classification can be as simple as either "orca", "voiceover" or "background", or can be as complex as classifying different call types or even different pods through their call repertoires.

One important design goal in this project is to make our system as flexible and extensible as possible. To this end, we have come up with a tag-based system that allows researchers to construct their own ontologies. Some of these ontologies would overlap, like "orca" or "voiceover", and some would be distinct to different areas of study. By allowing a user defined structure to emerge, we empower individual research communities to ask and answer the questions most pertinent to them. The obvious drawback to this is that within communities researchers must agree on the same language and syntax to annotate calls. We hope to provide tools within and external to the site to help encourage the collaboration necessary to converge on the same vocabulary.

3.6 Distributed Computing

In order to analyze the audio in large online archives efficiently, a scenario involving distributed computing must be used. There are many different types of systems that perform Machine Learning and many of these can distribute their work to different machines. Some of these systems are better suited for this task than others, and in this thesis we investigate a few combinations of these tools.

One of the possible solutions to the problem of the segmentation of audio into orca vocalizations and other sounds is to use a Machine Learning supervised classifier system. These systems require hand labeled training data, in our particular case, audio would be labeled as either orca or background. This training data is then used to train a classifier system that can classify future instances based on features of the training data. There are a wide variety of classification algorithms that are commonly used, some of the most popular are Support Vector Machines (SVMs), Neural Nets, Decision Trees and Logistic Regression. In our past work we have found good success in using SVMs, and have shown that SVMs typically outperform many other Machine Learning algorithms [NWMT08]. However, there is current work in this area showing the benefits of many alternate algorithms for different problem domains.

The second step is to take these audio features, train and test a Machine Learning classifier on them, and then run this Machine Learning classifier over all the data. In many cases it is beneficial to pre-calculate the audio features for the data, as once as good parameters can be determined for the audio feature extraction engine, such as the window size, hop size for the spectrum calculation and the memory size for the windowed statistical calculations, these features can be saved and reused in multiple Machine Learning experiments with different parameters. One example of this is the recent release of the Echonest audio features for the Million Song Dataset [BMEWL11]. This collection of audio features for one million popular songs took a considerable amount of time to process, and one approach is to save these audio features and running different machine learning algorithms or different parameter sets of these algorithms on them. Another point that should be made here is that the form of the features that are extracted in this step can have a direct relation to the performance of the subsequent Machine Learning step. Different algorithms perform better with different forms of data, and matching the right features to the right algorithm is both an art and a science.

Chapter 4

Case Study

4.1 Introduction

There are a number of large bioacoustic archives that have been described in the literature, and as computer storage grows ever larger, more of these archives are being created. In this work, we describe a large bioacoustic dataset, The Orchive which is an archive of sounds collected by a network of stationary, permanently mounted hydrophones, and the primary sound that was of interest to the researchers was that of Orcas.

These tools include a web-based interface to allow researchers to listen to, view and annotate this audio, tools to extract and view audio features from this data, and tools to run machine learning software and view the results of this analysis.

4.2 The Orchive

The whale species *Orcinus orca*, commonly known as Killer Whales [JFB00], are large toothed whales found around the world, in places as far afield as Antarctica and Alaska[EDSW09]. The vocalizations of orcas are complex and diverse, and consist of a wide variety of vocalizations, which include echolocation clicks, tonal whistles and pulsed calls [DFS00]. There are stable resident populations of Orcinus Orca in the northwest Pacific Ocean, some of these populations are found near Hanson Island, off the north tip of Vancouver Island in Canada. There are three distinct types of Orcas, Transients, Residents and Offshores, each of which have different feeding behaviors and different styles of communication.

Orcalab is a research station that has been recording audio of these Orca populations since 1972 [WLSS06]. It was designed as a land based research station in order to reduce the impact of the researchers on the orcas under study, as the noise and disturbance from boats impacts the orcas in observable but currently unquantified ways.

They have amassed a huge archive of more than 20,000 hours of audio recordings collected via a permanent installation of underwater hydrophones. It contains a continuous archive of orca vocalizations from the 1980 to the present time. This archive of data is very large, and the sound files that it contains occupy approximately 15TB of disk space. The archive was recorded onto cassette and DAT tapes.

The vocalizations of orcas are recorded through a series of hydrophones, microphones that are designed specifically to work under the water. A few of the hydrophones are hardwired, but in order to have greater geographical reach, most of the hydrophones operate remotely, transmitting their recordings via VHF radio to the main lab. These signals are then fed into a 6-channel mixing console. The researchers continuously adjust the levels of the recordings both to increase the volume of the hydrophone located closest to the orcas, and also to decrease the volume of any hydrophones that are close to sources of boat noise.

Our lab here at the University of Victoria has been collaborating with OrcaLab for approximately 6 years, and in this collaboration we have just recently completed the process of digitizing the collection of analog and DAT tapes.

Although these recordings contain large amounts of Orca vocalizations, the recordings also contain other sources of audio, including voice-overs describing the current observing conditions, boat and cruise-ship noise, and large sections of silence. Finding the Orca vocalizations on these tapes is a labor-intensive and time-consuming task.

Many parts of the recordings contain boat noise, which makes identifying orca calls both difficult and tiring. In addition, the size of the Orchive makes full human annotation practically impossible. Therefore, we have explored machine learning approaches to the task. One data mining task is to segment and label the recordings with the labels background, orca, voice. Another is to subsequently classify the orca calls into the classes specified in the call catalog. Experiments involving these two classification tasks will be explored in the Evaluation chapter of this thesis.

The goal of the Orchive project is to digitize acoustic data that have been collected over a period of 36 years using a variety of analog media at the research station OrcaLab (`http://www.orcalab.org`) on Hanson Island on the west coast of Vancou-

ver Island in Canada. Currently, they have approximately 20,000 hours of analog recordings, mostly in high quality audio cassettes. In addition to the digitization effort which is underway, we are developing algorithms and software tools to facilitate access and retrieval for this large audio collection. The size of this collection makes access and retrieval especially challenging (for example it would take approximately 2.2 years of continuous listening to cover the entire archive). Therefore the developed algorithms and tools are essential for effective long term studies employing acoustic techniques. Currently such studies require enormous effort as the relevant acoustic tapes need to be recovered and the relevant segments need to be tediously digitized for analysis.

The majority of the audio recordings consist of three broad classes of audio signals: background noise caused mainly by the hydrophones, boats, background noise containing orca vocalizations and voice over sections where the observer that started the recording is talking about the details of the particular recording. In some cases there is also significant overlap between multiple orca vocalizations. The orca vocalizations frequently can be categorized into discrete calls [For89] [For91] that allow expert researchers to identify their social group (matriline and pod) and in some cases even allow identification of individuals.

Even when the data is digitized, locating a particular segment of interest in a long monolithic audio recording can be very tedious as users have to listen to many irrelevant sections until they can locate what they are looking for. Even though visualizations such as spectrograms can provide some assistance, this is still a task that requires much manual effort. In this thesis, we describe experiments for the automatic classification and segmentation of the orca recordings for the purposes of locating segments of interest and facilitating interaction with this large audio archive.

An important aspect in the design of a tool to support collaborative work is to consider what user communities will use the tool. In the case of the Orchive, there are a number of different scientific communities that will be using this tool and the data this tool provides access to. The primary scientific community that will benefit from this work will be cetacean biologists. In order to study the rich archive orca vocalizations that have been recorded by Orcalab, researchers must currently travel to Hanson Island, search through the lab books and incidence reports to find which recordings contain the data they are interested in, locate the physical cassette tape corresponding to this recording, and then either manually listen to the tape, or perhaps digitize the tape and analyze it in the computer. Each researcher typically then

keeps the annotations and data generated from this procedure themselves, if future researchers want to obtain this data for further analysis, they must first be aware of the fact that this researcher has the data, and then request it from them. With the distributed collaborative system we have designed, not only can these biologists easily listen to any recording in the entire archive from any internet connected computer in the world, and compare different recordings, they can also add their annotations to the system. These annotations can be either private or public, if they are for use in a publication, after the article has been accepted for publication, the researcher can make their private annotations public. These researchers are less interested in the details of audio feature extraction and machine learning algorithms, and are instead more focused on asking biologically informed questions, like dialect change in cetacean call repertoire [DFS00].

Another scientific community that will receive benefits from this archive are the developers of bioacoustic algorithms. These scientists are typically computer scientists with interests in Music Information Retrieval and bioacoustics. This archive represents a site where researchers can get large amounts of high quality and uniformly collected data. Researchers interested in bioacoustic algorithms have different goals and skill sets from cetacean biologists, for example, many have extensive knowledge of Digital Signal Processing and audio feature extraction algorithms. This system should be flexible and powerful enough to allow these researchers to ask questions that are relevant to them. The required features for this group of users include allowing them to choose different audio feature extraction algorithms, and to then take the resulting data and run it against a variety of machine learning algorithms in as flexible a manner as possible.

In this work we design a series of Machine Learning systems to examine and classify various aspects of this archive. We first look at classifying different recordings based on how much noise is present in each recording. There is often considerable boat noise in the recordings, and in many cases this makes analysis difficult or unfeasible. By identifying recordings with less boat noise, we can narrow the amount of data that is required to be analyzed by humans and machine learning systems.

The second task in this work is to develop machine learning systems to pull out individual orca calls from recordings. These systems should discriminate between background noise, orca calls and the voice notes that are present in most of the tapes in the Orchive. Once we have identified clips containing orca vocalizations, the third task in this work is to classify these calls based on a previously existing call catalog,

first developed by John Ford.

Over the past 5 years, a number of orca researchers using our website have added over 18,000 clip annotations to our database. A small section of annotated audio from the Orchive is shown in Figure 3.2. These clip annotation are of two main types: The first is clips that differentiate background noise from orca calls and from the voice notes of the researchers that collected the data. The second type of clip annotations classify orca vocalizations into different calls. Orcas make three types of vocalizations, echolocation clicks, whistles and pulsed calls. The pulsed calls are highly conserved stereotyped vocalizations which have been classified into a catalog of over 52 different calls by John Ford [For87]. Of the 18,000 annotations currently in the Orchive, 3000 are of these individually classified calls. In addition, we have a curated call catalog containing 384 different recordings of different calls vocalized by a variety of different pods and matrilines. This catalog is used for training the annotators.

Because the researchers from Orcalab have devoted so much time and care to the recording and documenting the audio in the Orchive, an important design criteria is to ensure that data entered from non-specialist users will not interfere with the data from expert users and scientists. To this end, we have implemented a reputation management and filtering system, along with roles for various users.

In the current implementation of the Orchive, only expert scientists are allowed to make annotations, and only members of Orcalab and their collaborators are allowed to edit the start date of each recording. As well, users can choose exactly which user's annotations they want to view. In light of the fact that everyone in the current user community knows each other personally, reputation management is handled by offline personal interactions. When the general public is allowed to contribute using a crowdsourcing model, this system will be expanded and enhanced.

Chapter 5

Evaluation

The final goal of the work in this thesis is to take the raw audio obtained from large scale recording projects and to find and label the audio events made by the organisms that are being studied. In the case of the Orchive, these are in general the various vocalizations of orcas, and more specifically are the stereotyped pulsed calls. For this project, the final goal would be to assign call types to each vocalization from the call catalog of Ford [For87], and to label the clan, pod, subpod, matrilinal association of the orca making these calls. The difficulty of assigning each call to a group of whales becomes more difficult as more specific groups of whales are considered, and varies considerably for each call type. For example the N47 call is made exclusively by the A1 pod of whales, so if this call is detected, the identity of the pod making this vocalization can be assigned at the same time. In addition, there is considerable variation in the production of the N47 by different matrilines (A12, A30, A34, A36), so the assignment of matrilinal identity to this call could also be possible. Other calls, like the N03 call, are made by all the pods in the A clan (A1, A4, A5, B, C, D, H, I1) and has little variation between the different pods, so assigning this call to a specific pod or matriline is difficult.

The process of classification of audio involves three major steps, the first being the labeling of audio with ground truth by experts, audio feature extraction of audio, and the use of machine learning to classify the audio features with the ground truth labels from experts.

For the labeling of audio, a web based system was developed for this thesis that allowed experts to listen to, view and label audio data. This system allowed for the use of very large corpora of audio, with long audio files of 45 minutes in length and greater.

For audio feature extraction, the primary toolkit that was used was the Marsyas [Tza08] framework, an extensible system based on a data flow metaphor containing subsystems that allowed for the computation of many different audio features. It also contained low level subsystems that allowed for new audio features to be quickly implemented.

In the course of work done for this thesis, this author added new audio feature extraction subsystems to Marsyas, including the porting of the Yin pitch detector from Aubio to Marsyas

For machine learning, a variety of different machine learning systems were used. One of these was libSVM [CL01] which was used for doing general Support Vector Machine (SVM) training and prediction using a variety of kernels. The LIBLINEAR [FCH+08] package was used for doing SVM classification using linear kernels, and contains code that is able to greatly speed up SVM classification by taking advantage of the linear hyperplanes used in linear SVM kernels. The Weka machine learning package [WF05] was used to test a variety of different machine learning algorithms including decision tree classifiers, random forests of decision trees, naive Bayes classifiers, and simple perceptron based models.

5.1 Orca/Background/Voice Segmentation

The first step in classifying vocalizations of organisms from large recordings is to segment the recording into sections containing the species of interest from background noise and other sounds. In the case of the Orchive, there are three primary sources of sound on the tapes. The first is the vocalizations of orcas, the second is background noise, and the third is a voice note made by the researchers at OrcaLab with information about the date, time, tape number, audio mixer parameters and other information about the recording. In this work we refer to this as the Orca/Background/Voice task, and to the dataset that was used as ORCAOBV.

For this task, we employed two different methods, the first was to ask experts in orca calls to label recordings based on if they contained orca calls, background, or voice, using a web based interface written in Flash. Over the course of 5 years we recruited 12 experts to help with this project, and obtained 9866 annotations of orca, 1829 of background and 453 of voice. This represented a considerable amount of work for the experts, but only represented a total of 360 minutes of orca calls, 72 minutes of background and 93 minutes of voice, for a total or 8.75 hours of annotations out

of the total 20,000 hours of the Orchive.

In order to leverage the work done by these experts, we used their annotations as the input to an audio feature extraction and machine learning system that was developed specifically for this thesis, one that used several widely used audio feature extraction and machine learning toolkits described in the introduction.

In the following sections, the results of different sets of audio features with different machine learning algorithms are shown.

Mel-Frequency Cepstral Coefficients (MFCCs) are a feature that has shown great utility in the classification of human speech and of music signals. It is a cepstral representation of audio, which can be thought of as a spectrum of a spectrum, with the frequency bands spaced at the mel scale, a scale that approximates human hearing. Variable numbers of coefficients can be calculated with the MFCC, typically 13 are used, but more have also been used in a variety of studies. The Marsyas audio framework provides a MFCC subsystem, and all the results here use Marsyas to calculate the MFCC coefficients.

Mel-Frequency Cepstral Coefficients [Log00] (MFCC) have been widely used for this purpose. MFCCs have also been used in bioacoustics, and have been used to classify insect sounds [LQ11], birds [LSYS07] and orca calls [NWMT08]. We have investigated the use of MFCC values for classifying calls from the orca call catalog, and results using these with a variety of machine learning techniques are described below. In future work, we would like to try to use MFCC or other forms of spectral data as input to our symbolic approximation algorithm.

MFCC features are derived from a power spectrum of the features, which takes as parameters the window size and hop size. Another parameter is the number of windows over which to take the mean and standard deviation of the MFCC coefficients. We did a parameter scan over these quantities to find the optimal values for these parameters. A diagram showing the definition of window size, hop size, memory 5.1.

You can also change the number of MFCCs, 13 is a typical number, 20 is also used. All these experiments were done with a window size of 10 frames. The number of MFCCs that can be calculated is related to the window size of the FFT, and for smaller window sizes, the larger number of MFCCs couldn't be calculated. For example, at a window size of 512 samples, a maximum of 40 MFCC components could be calculated. Results of this scan are shown in Table 5.2.

From this, one can see that in general the more MFCC coefficients that are calculated, the better classification accuracy is obtained. Again as in the table where

ws	hp	mem	Extract Time	Train Time	Predict time	Percent Correct
512	256	1	373.28	55.62	1.14	74.25
512	256	2	401.60	300.45	2.14	75.57
512	256	5	351.78	219.87	2.05	75.70
512	256	10	356.74	212.28	2.07	75.89
512	256	20	408.21	236.77	2.33	76.15
512	256	40	342.78	185.42	2.10	76.56
512	256	80	432.55	168.96	2.06	76.97
512	256	160	489.03	162.82	2.06	76.73
1024	512	1	338.41	21.12	0.59	74.10
1024	512	2	308.87	124.84	1.05	75.33
1024	512	5	351.13	89.28	1.07	75.50
1024	512	10	316.43	88.40	1.06	75.74
1024	512	20	290.46	99.90	1.03	76.16
1024	512	40	342.48	78.47	1.06	76.71
1024	512	80	254.50	79.73	1.06	76.36
1024	512	160	271.35	112.03	1.09	76.38
2048	1024	1	283.75	10.24	0.30	73.74
2048	1024	2	236.16	49.18	0.55	75.47
2048	1024	5	262.77	51.33	0.55	75.93
2048	1024	10	245.09	43.77	0.55	75.90
2048	1024	20	254.97	44.08	0.57	76.46
2048	1024	40	282.02	51.35	0.54	76.54
2048	1024	80	254.46	40.00	0.52	76.38
2048	1024	160	271.33	42.40	0.61	74.82
4096	2048	1	251.30	4.89	0.23	73.53
4096	2048	10	245.09	19.91	0.29	75.94
4096	2048	160	271.34	23.71	0.29	74.25
4096	2048	2	236.16	25.66	0.27	75.69
4096	2048	20	262.91	20.60	0.27	75.93
4096	2048	40	261.24	19.96	0.26	76.31
4096	2048	5	262.77	21.16	0.28	75.93
4096	2048	80	254.44	20.94	0.28	74.79

Table 5.1: Table of MFCC results with different window sizes using the LIBLINEAR classifier. In this and subsequent tables, “ws” refers to the size of the FFT window in samples, “hp” refers the hop size between sequent FFT frames in samples, and “mem” refers to the size of the texture window in frames.

Figure 5.1: A graphical representation of window size, hop size and memory size in the Audio Feature Extraction algorithm that has been used.

window size, hop size and memory were calculated, the size of the window that is used for the FFT determines the size of the classification frames, and at a sampling rate of 44100 samples per second, and with an texture window of 10 frames, a hop size of 2048 samples would take information from a duration 0.464 seconds, where a hop size of 1024 samples would be 0.232 seconds.

Because for segmentation it is important to not include excessive amounts of silence before and after the call, it was decided to use a window size of 2048 samples, which gave a classification accuracy of 84.83% rather than the slightly higher performing hop size of 2048 samples, which increases the classification accuracy by 1.33%. For this reason, subsequent tables are performed with a window size of 2048 samples and a hop size of 1024 samples.

LIBLINEAR can run two different binary linear classifiers, Logistic Regression (LR) and a linear Support Vector Machine (SVM) [FCH+08] If one has a set of instances with labels:

ws	hs	Num Mfccs	Extract Time	Train Time	Predict time	Percent Correct
512	256	1	268.03	34.37	0.28	74.21
512	256	5	458.40	112.20	0.90	75.55
512	256	10	482.89	216.13	1.67	75.82
512	256	15	544.28	307.55	2.61	76.03
512	256	25	604.75	516.71	5.65	76.48
512	256	30	637.92	627.21	4.66	76.52
512	256	35	584.31	815.73	5.43	76.56
512	256	40	589.18	828.87	9.29	76.59
1024	512	1	319.74	10.21	0.19	74.21
1024	512	5	380.35	37.84	0.44	75.61
1024	512	10	369.72	82.38	0.87	75.72
1024	512	15	400.81	136.09	1.28	75.82
1024	512	25	397.25	279.25	1.94	76.08
1024	512	30	397.25	322.54	2.36	76.35
1024	512	50	474.41	678.87	3.85	79.81
1024	512	60	477.28	755.53	5.21	79.77
1024	512	70	532.27	1098.05	5.94	79.78
1024	512	80	531.63	1317.27	7.39	79.82
2048	1024	1	276.15	5.93	0.09	74.24
2048	1024	5	280.41	18.35	0.26	75.28
2048	1024	10	291.82	31.22	0.45	75.71
2048	1024	25	320.04	106.98	1.00	78.07
2048	1024	30	347.35	192.40	1.23	78.45
2048	1024	50	355.68	292.04	1.93	84.81
2048	1024	60	394.41	392.25	2.63	84.81
2048	1024	70	394.40	482.52	2.75	84.82
2048	1024	80	414.39	646.53	4.65	84.83
2048	1024	90	421.30	762.10	3.43	84.84
2048	1024	100	421.29	789.52	5.84	84.83
4096	2048	1	283.65	2.00	0.06	74.28
4096	2048	5	282.06	9.25	0.18	74.93
4096	2048	6	272.92	10.24	0.17	75.13
4096	2048	10	271.01	18.92	0.23	75.57
4096	2048	15	235.32	26.94	0.32	76.99
4096	2048	25	282.06	71.18	0.50	79.44
4096	2048	30	269.92	74.77	0.60	79.93
4096	2048	50	275.42	130.27	1.02	86.19
4096	2048	60	292.83	172.26	1.16	86.18
4096	2048	70	328.59	218.28	1.35	86.09
4096	2048	80	325.09	233.88	1.51	86.16
4096	2048	90	339.80	308.25	1.72	86.14
4096	2048	100	388.68	469.48	1.98	86.17

Table 5.2: Table of MFCC results with different numbers of MFCC coefficients using the LIBLINEAR classifier.

$$(x_i, y_i), i = 1, ..., l x_i \in R^n, y_i \in -1, +1$$

Both methods solve the following optimization problem:

$$min_w 1/2 w^T w + C \sum_{i=1}^{l} \xi(w; x_i, y_i)$$

where C is a penalty parameter. The SVM can employ two different loss functions, a L1 loss function $max(1 - y_i w^T x_i, 0)$ and an L2 loss function $max(1 - y_i w^T x_i, 0)^2$. In the case of linear regression, the loss function is $log(1 + e^{-y_i w^T x_i})$.

LIBLINEAR has 8 different solvers that can be used in multi-class classification, these different solvers solve different forms of the function, and either solve the primal problem or the dual problem. For every primal form of a function, there exists a dual form which gives the maximum upper bound of the problem, and is sometimes quicker to solve than the primal form of the problem, especially in the case of linear SVM. In addition, the solver can be L1 regularized or L2 regularized. Finally, the loss term can be either an L1 loss or an L2 loss.

It has been shown [FCH+08] that these different solvers can yield small differences in classification accuracies and can show large changes in the amount of time that each takes to solve a given classification task.

Another important parameter to vary for LIBLINEAR is the penalty, or cost, parameter C. However, a theoretical proof has been presented [KL03] that for the solvers in LIBLINEAR, once C has reached a certain threshold the obtained models will have a similar performance.

The results from doing a parameter sweep over all solvers in LIBLINEAR and values of C from 0.001 to 1000 are presented in Table 5.4. In this table, the solver id is given, and a mapping between this id and the type of the solver is shown in Table 5.3

In this table, two different window sizes are used, a small window of 512 samples with a hop size of 256 samples, and a large window of 4096 samples with a hop size of 2048 samples.

From this, we can see that indeed the different solvers and different values of the cost parameter only have minor effects on the overall accuracy of the classifier. However, in terms of speed, there are dramatic differences between the classifiers, and in general the L2-regularized L2-loss support vector classification primal solver has

0	L2-regularized logistic regression (primal)
1	L2-regularized L2-loss support vector classification (dual)
2	L2-regularized L2-loss support vector classification (primal)
3	L2-regularized L1-loss support vector classification (dual)
4	support vector classification by Crammer and Singer
5	L1-regularized L2-loss support vector classification
6	L1-regularized logistic regression
7	L2-regularized logistic regression (dual)

Table 5.3: Table of LIBLINEAR parameters

ws	hp	Solver	C	Extract Time	Train Time	Predict time	Percent Correct
512	256	1	0.001	389.60	38.87	2.07	75.75
512	256	1	1.0	387.99	219.49	2.07	75.89
512	256	1	1000.0	386.40	2444.94	2.10	75.82
512	256	2	0.001	385.36	31.83	2.06	75.77
512	256	2	1.0	405.99	33.61	2.05	75.90
512	256	2	1000.0	405.82	31.86	2.12	75.90
512	256	3	0.001	404.96	33.51	2.04	75.44
512	256	3	1.0	406.80	262.06	2.08	75.77
512	256	3	1000.0	409.53	2459.79	2.07	75.70
512	256	4	0.001	409.54	29.51	2.05	75.76
512	256	4	1.0	415.15	280.75	2.09	75.79
512	256	4	1000.0	415.87	70845.55	4.92	75.77
512	256	5	0.001	414.78	132.72	2.05	75.77
512	256	5	1.0	416.51	174.16	2.05	75.91
512	256	5	1000.0	417.28	141.37	2.25	75.91
512	256	6	0.001	417.28	67.80	3.02	75.86
512	256	6	0.1	416.09	54.47	2.08	76.16
512	256	6	1000.0	429.14	72.80	2.07	76.15
512	256	7	0.001	429.14	61.21	2.05	75.12
512	256	7	1.0	435.12	135.04	2.09	76.17
512	256	7	1000.0	299.58	4537.57	2.38	75.32

Table 5.4: A table showing the results of doing a parameter search over a wide number of values of C and using the different solvers in the LIBLINEAR package. Please note that approximately 6x as many results as would fit here were also calculated, but were omitted due to space restrictions. These omitted results showed approximately the same behaviour as those above.

the most consistent speed of all the solvers. For this reason, all subsequent tables will use this solver, and will use a cost parameter of C.

Although this parameter search for LIBLINEAR gave similar classification accuracy for all conditions, this is highly problem dependent, and for different problems this parameter search can often yield better results, and many sources, including the original LIBLINEAR paper highlight the importance of doing parameter searches.

In the field of Music Information Retrieval, a wide variety of different audio features have been used successfully in the past to classify a variety of audio signals. In the next section, a number of these audio features are evaluated as for their performance in classifying recordings into the three classes of orca/background/voice.

A very simple feature that derives directly from the waveform of the signal is the Root Mean Square (RMS) energy of a signal. The RMS is one measure of the amount of energy present in a waveform. The RMS is calculated using the following formula:

$$\sqrt{1/N \sum_{i=0}^{N} x^2}$$

The power spectrum of a signal can be obtained by using a Fast Fourier Transform. This Fourier analysis of a signal decomposes a time sequence of a waveform into separate frequencies. This frequency representation of a signal can be represented pictorally as a spectrogram and is a widely used method for researchers to visually identify different bioacoustic sounds. Statistical properties of this spectrum can also be obtained, and these properties have been used previously in the Music Information Retrieval literature to classify audio signals.

In this thesis, we use five of these statistical measures of a signal, these are the Centroid, Rolloff, Flux, Kurtosis, Skewness of the spectrum [Tza08] [KD06].

The Spectral Centroid of a signal is defined as the normalized first moment, or center of gravity of a signal [LO06] and is represented by the following equation:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}$$

The Spectral Rolloff is the frequency below which 90% of the magnitude in frequency bins is concentrated and is defined by the following equation:

$$\sum_{n=1}^{R_t} M_t[n] = 0.90 * \sum_{n=1}^{N} M_t[n]$$

The Spectral Flux is the norm of two successive power spectrum and is shown in the following equation:

$$\sum_{n=1}^{N}(N_t[n] - N_{t-1}[n])^2$$

The Spectral Kurtosis of a signal is a measure of the peakedness of a signal and is described by the following equation [DT96]:

$$\frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^3}{(N-1)s^3}$$

A measure closely related to the kurtosis is the Spectral Skewness of a signal, and is defined by the following equation:

$$\frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^4}{(N-1)s^4}$$

These statistical properties of a signal can be used separately, however, when used in combination often give a better description of the spectrum.

Another feature often used in MIR is Chroma features. In this representation, the entire spectrum is projected onto 12 bins, each of which corresponds to one of the semitones of the equal tempered musical octave. Although this measure is of more use when analyzing Western Music, it could be useful if the vocalizations that are being studied have specific absolute pitches.

The Spectral Flatness Measure (SFM) [HK07] is a measure of the flatness of a spectrum. Under this measure white noise has a flat spectrum. This measure has been shown to be useful for discriminating voiced and un-voiced components in human speech. It is defined by the following equation:

$$SFM_{i,b} = \frac{[\prod_{f=l_b}^{u_b} |S_i[f]|^2]^{\frac{1}{u_b - l_b + 1}}}{\frac{1}{u_b - l_b + 1}\sum_{f=l_b} u_b |S_i[f]|^2}$$

The Spectral Crest Factor (SCF) provides a quantified measure for the tonality of a signal, and indicates the relative peak of a subband which has been shown to be useful [HK07] for discriminating wideband and narrowband signals. Each peak in the output of this formula gives the dominant pitch frequency in each subband. It is defined by the following equation:

$$SCF_{i,b} = \frac{max(|S_i[f]|^2)}{\frac{1}{u_b - l_b + 1}\sum_{f=l_b}^{u_b} |S_i[f]|^2}$$

A complementary approach to spectral methods for analyzing a signal are autocorrelation based methods, which are often used is an estimate of the fundamental frequency (F0) or pitch. One such well known algorithm for this is the Yin algorithm [dCK02]. This algorithm is primarily an autocorrelation based approach, which means that it takes the audio signal and convolves it with itself in the time domain. The peaks in this convolution then correspond to harmonics in the signal, and with noise free data with harmonics that strictly decrease the lowest peak is the fundamental frequency. With audio that does not fit this strict definition, there are many cases where the lowest peak is not the fundamental frequency, one example is if odd harmonics are systematically lower than even harmonics, and another is if there is substantial noise in the data. The Yin algorithm makes several modifications to simple autocorrelation to overcome these issues.

The discrete calls of killer whales are pulsed signals in which a tone (of a certain tonal frequency) is not emitted continuously but in pulses given by the pulse-repetition rate. Unlike the tonal signals of many birds and other delphinids, the highest energy is not always contained in the first or second harmonic [VDS99]. This causes problems for autocorrelation based methods such as Yin, and often integer multiples of the fundamental frequency are found rather than the true fundamental. These errors are commonly called octave errors. Visual inspection of the output of the Yin algorithm in the Orchive Version 2.0 website shows that these octave errors are common in some recordings, but in some recordings these errors are seen less.

In addition, the presence of noise can prevent this algorithm from finding the fundamental frequency of the orca call. In addition, if the noise has harmonic structure the YIN algorithm can find the pitch of the boat noise. This is visible in a number of the Orchive recordings.

To investigate the utility of using these different features to classify the recordings in the Orchive, these features were calculated for the same set of recordings in the previous tables. These features were first evaluated separately, and then were joined together into various combinations, which were also used as input to the LIBLINEAR algorithm with the L2-regularized L2-loss support vector classification (primal) solver and a cost parameter of 1.0.

In Table 5.5 the results of using MFCC coefficients on their own is shown. The

highest classification accuracy obtained is for a window size of 256 and a hop size of 128 with a classification accuracy of 76.46%. However, the difference between the results was only 0.7%, and for each halving in hop size, the number of feature vectors output are doubled. For large datasets like the 1GB dataset of audio in the ORCAOBV dataset, this can lead to very large sets of feature vectors, which leads to longer extraction, training and prediction times, which is evident in this table.

The results of using RMS features is shown in Table 5.6, and surprisingly shows that simply looking at the energy of the signal gives quite good performance, but with lower performance than that of using MFCC features. This is likely because recordings that have orca vocalizations plus boat noise are louder than recordings that contain only boat noise. The high classification accuracy using RMS is likely a result of the way that the dataset in this thesis was constructed. The procedure that was followed was that each 100th recording in the Orchive was examined by hand, and one or several examples of orca calls in this recording was annotated. Directly around the vocalization, regions of background noise were selected. Because of this, the SVM algorithm had at its disposal examples of recordings containing boat noise plus orca as well as examples of this same boat noise without orca. In general, one could not guarantee that datasets would contain this structure of clips, and performance would likely be lower when using RMS alone in these cases.

In Table 5.7 the results of using different statistical measures of the signal including the Spectral Centroid, Rolloff, Flux, Kurtosis and Skewness are shown. In general, these features perform fairly well on their own, with classification accuracy being around 73%-76%. When the Centroid, Rolloff and Flux are combined into one measure, performance improves to a maximum of 80.51% for a window size of 4096 samples and a hop size of 2048 samples. This improved performance of a combination of spectral statistical features performing better than a single feature has been seen in the literature [Tza08].

The results of using chroma features is shown in Table 5.8, and shows good performance, with a peak of 79.47% classification accuracy. In this same table results are shown for using the Spectral Crest Factor (SCF) which gives a maximum performance of 84.25%. The Spectral Flatness Measure gives a maximum performance of 86.23% and the combination of all three gives a maximum performance of 88.96%. In this table the large feature file size created by a hop size of 128 and 256 samples led to the output of a file greater than 3GB in size, which was larger than the maximum default size of input file capable of being loaded into LIBLINEAR without changes to the

ws	hp	Extract Time	Train Time	Predict time	Percent Correct
256	128	594.27	74.61	4.32	76.46
512	256	358.32	33.11	2.04	75.90
1024	512	269.96	15.89	1.02	75.76
2048	1024	222.55	8.64	0.55	75.92
4096	2048	209.66	4.06	0.28	75.93

Table 5.5: Table showing the classification accuracy and timing when using different FFT window sizes on MFCC features using a linear SVM kernel.

ws	hp	Extract Time	Train Time	Predict time	Percent Correct
256	128	352.13	6.55	0.54	74.21
512	256	273.31	3.08	0.27	74.17
1024	512	264.65	1.79	0.16	74.13
2048	1024	194.95	0.83	0.07	74.20
4096	2048	193.86	0.45	0.06	74.24

Table 5.6: Table showing the effect of window size on classification performance with LIBLINEAR using the Root Mean Square (RMS) energy of a signal as a feature.

source code, this led to the failure of the training step, and in this case the accuracy is denoted by -1.0. With the addition of more features, the size of the feature file must be kept in mind, and a tradeoff between more feature vectors and performance accuracy must be made. In this case, a window size of 2048 and a hop size of 1024 gave a reasonably sized feature file and also provided the highest performance.

When used on its own, the YIN algorithm showed acceptable performance, with an maximum classification accuracy of 74.69% as is shown in Table 5.9. However, when combined with other features, as is shown in table 5.10 performance improved and reached a maximum of 89.57% when all the features mentioned above were used with a window size of 2048 and a hop size of 1024. This represented the best results observed when using the LIBLINEAR SVM classifier.

One way to reduce the total number of feature vectors is to downsample the feature vectors at the time that features are calculated. In this case, downsampling is done by only outputting one feature vector each N times the feature is calculated. In table 5.11 results are shown of using window sizes from 512 to 4096 and downsampling rates from 1, where every feature vector is output, to 1000, where only every 1000th feature vector is used. What one would expect in this table is that a downsampling of 1 would give the highest accuracy, and greater amounts of downsampling should either the same accuracy until too few feature vectors are used and performance would

Extractor	ws	hp	Extract Time	Train Time	Predict time	Percent Correct
centroid	256	128	417.19	8.15	0.62	74.21
centroid	512	256	277.15	3.71	0.27	74.18
centroid	1024	512	269.42	1.79	0.14	74.16
centroid	2048	1024	221.64	0.95	0.07	74.20
centroid	4096	2048	209.47	0.46	0.04	74.24
flux	256	128	489.00	7.04	0.64	74.22
flux	512	256	276.48	3.00	0.27	74.05
flux	1024	512	268.81	1.78	0.14	74.11
flux	2048	1024	220.77	0.83	0.08	74.24
flux	4096	2048	208.09	0.43	0.04	74.28
kurtosis	256	128	411.39	6.79	0.56	73.01
kurtosis	512	256	275.30	3.30	0.28	71.58
kurtosis	1024	512	268.23	1.90	0.14	70.32
kurtosis	2048	1024	219.85	0.83	0.10	69.30
kurtosis	4096	2048	207.49	0.46	0.06	68.96
rolloff	256	128	414.58	6.90	0.58	75.05
rolloff	512	256	275.80	3.31	0.27	74.47
rolloff	1024	512	268.86	1.78	0.14	74.53
rolloff	2048	1024	221.24	0.83	0.07	74.60
rolloff	4096	2048	209.40	0.42	0.04	74.44
skewness	1024	512	464.53	1.74	0.14	73.57
skewness	2048	1024	771.85	0.93	0.07	71.10
skewness	256	128	418.48	6.55	0.54	74.22
skewness	4096	2048	1294.41	0.44	0.04	64.83
skewness	512	256	368.13	3.43	0.61	74.11
all	256	128	659.20	91.27	5.10	76.15
all	512	256	442.53	45.34	2.51	75.92
all	1024	512	306.38	24.25	1.25	76.75
all	2048	1024	216.53	9.79	0.65	78.84
all	4096	2048	210.16	5.06	0.32	80.51

Table 5.7: Table showing the effect of using combinations of different statistical measures of the spectrum of a signal on classification performance with LIBLINEAR.

Options	Extract Time	Train Time	Predict time	Percent Correct
chroma -ws 256 -hp 128	573.98	73.88	4.25	75.11
chroma -ws 512 -hp 256	364.39	35.76	2.12	77.65
chroma -ws 1024 -hp 512	270.59	21.38	1.18	79.47
chroma -ws 2048 -hp 1024	258.78	11.64	0.56	78.91
chroma -ws 4096 -hp 2048	230.19	5.01	0.27	77.14
scf -ws 256 -hp 128	661.55	0.02	0.02	NC
scf -ws 512 -hp 256	378.20	61.65	3.74	81.58
scf -ws 1024 -hp 512	267.59	34.72	2.00	83.69
scf -ws 2048 -hp 1024	218.75	16.25	0.92	84.25
scf -ws 4096 -hp 2048	230.07	10.27	1.31	83.56
sfm -ws 256 -hp 128	759.73	134.96	10.27	79.06
sfm -ws 512 -hp 256	430.28	57.40	3.64	82.03
sfm -ws 1024 -hp 512	300.40	31.29	1.92	84.94
sfm -ws 2048 -hp 1024	298.46	13.87	0.92	86.23
sfm -ws 4096 -hp 2048	207.19	8.27	0.48	85.30
all -ws 256 -hp 128	1182.81	0.04	0.00	NC
all -ws 512 -hp 256	752.27	0.01	0.00	NC
all -ws 1024 -hp 512	558.47	99.95	5.27	88.02
all -ws 2048 -hp 1024	377.57	48.91	2.28	88.96
all -ws 4096 -hp 2048	319.81	22.95	1.18	87.79

Table 5.8: Table showing the classification performance of LIBLINEAR with different combinations of Chroma, Spectral Crest Factor (SCF) and Spectral Flatness Measure (SFM). Locations in the table with NC represent "Not Completed" combinations where the size of the resulting feature file was greater than the maximum 3GB file size allowed by LIBLINEAR.

Options	Extract Time	Train Time	Predict time	Percent Correct
yin -ws 256 -hp 128	434.67	5.76	0.47	74.71
yin -ws 512 -hp 256	363.36	3.18	0.24	74.69
yin -ws 1024 -hp 512	461.60	1.50	0.13	74.66
yin -ws 2048 -hp 1024	719.33	0.71	0.09	74.63
yin -ws 4096 -hp 2048	1218.52	0.37	0.03	74.57

Table 5.9: Table showing the result of using the YIN pitch estimator as a feature for input to the LIBLINEAR SVM classifier with different window sizes and hop sizes.

Options	Extract Time	Train Time	Predict time	Percent Correct
spectral+sfm+scf -ws 1024 -hp 512	777.38	91.02	4.89	88.23
spectral+sfm+scf -ws 2048 -hp 1024	963.55	43.61	2.44	88.96
spectral+sfm+scf -ws 256 -hp 128	1364.61	0.02	0.00	NC
spectral+sfm+scf -ws 4096 -hp 2048	1510.14	22.28	1.24	88.22
spectral+sfm+scf -ws 512 -hp 256	931.93	0.02	0.01	NC
spectral+css -ws 256 -hp 128	1615.31	0.05	0.00	NC
spectral+css -ws 512 -hp 256	1003.44	0.01	0.00	NC
spectral+css -ws 1024 -hp 512	884.71	109.31	5.89	88.96
spectral+css -ws 2048 -hp 1024	1040.40	55.77	3.01	89.53
spectral+css -ws 4096 -hp 2048	1557.60	31.19	1.59	88.17
spectral+css+yin -ws 256 -hp 128	1824.78	0.03	0.00	NC
spectral+css+yin -ws 512 -hp 256	1173.49	0.02	0.02	NC
spectral+css+yin -ws 1024 -hp 512	1149.11	117.97	5.97	89.02
spectral+css+yin -ws 2048 -hp 1024	1555.98	54.77	3.01	89.46
spectral+css+yin -ws 4096 -hp 2048	2648.89	29.90	1.54	88.12
spectral+css+yin+rms -ws 256 -hp 128	1824.54	0.03	0.00	NC
spectral+css+yin+rms -ws 512 -hp 256	1173.07	0.02	0.02	NC
spectral+css+yin+rms -ws 1024 -hp 512	1148.03	114.72	6.11	88.92
spectral+css+yin+rms -ws 2048 -hp 1024	1555.78	59.51	3.01	89.57
spectral+css+yin+rms -ws 4096 -hp 2048	2648.28	32.07	1.53	88.17

Table 5.10: Table showing the use of all audio features described above as input to the LIBLINEAR package. In this table "css" refers to the combination of Chroma, SCF and SFM features.

Window Size	Hop Size	Downsampling	Train Time	Predict time	Percent Correct
512	256	1	31.27	2.06	75.90
512	256	10	3.26	0.21	76.01
512	256	100	0.31	0.03	75.82
512	256	1000	0.07	0.02	75.00
1024	512	1	17.80	1.05	75.76
1024	512	10	1.64	0.12	75.76
1024	512	100	0.15	0.02	76.52
1024	512	1000	0.02	0.00	77.67
2048	1024	1	8.60	0.52	75.92
2048	1024	10	0.86	0.06	75.96
2048	1024	100	0.08	0.01	74.90
2048	1024	1000	0.01	0.00	73.08
4096	2048	1	3.96	0.30	75.93
4096	2048	10	0.36	0.04	75.89
4096	2048	100	0.09	0.01	76.54
4096	2048	1000	0.03	0.02	57.69

Table 5.11: Table showing the results of different amounts of downsampling on classification performance with the LIBLINEAR SVM using MFCC audio features. Please note that approximately 3x as much data points as this were calculated but were omitted due to space. These omitted data points showed the same approximate performance as those shown here.

decrease. In this table, the accuracy is quite similar inside a single window size, for example, for the case of a window size of 512, the difference in accuracies is 1.21%. This tells us that in general, with the large amount of data that is in this training set, any amount of downsampling will give approximately the same results. This could be used to great effectiveness when expanding the set of the training data to a larger size than the 11,041 clips used in this section as is shown by the dramatic reduction in training and test times that are seen in Table 5.11.

However, in some cases, higher accuracies for larger downsampling rates is seen. Upon further investigation, this turned out to be due to the fact that many clips were of short duration, with many being approximately 0.2 seconds in length. At this resolution, all of the feature vectors for some of these short recordings were discarded, and they therefore did not become part of either the training or test set. For this reason, all subsequent tables do not use downsampling, and to accommodate the long training and prediction times, a large cluster was used, and each job was given the maximum allowable three day run time to complete.

Like LIBLINEAR, the LibSVM [CL01] package allows researchers to train a Support Vector Machine, and was written by the same team that wrote LIBLINEAR. However, LIBLINEAR only allows for the use of a linear kernel, where LibSVM allows for the use of different kernels, including linear, polynomial, sigmoid and Radial Basis Function (RBF) kernels. For problems that have data points that are linearly seperable, a linear kernel works well, but for problems where the data points have more complex structure, like in optical character recognition, the choice of an appropriate kernel can project these points into a higher dimensional space, where these points can be separated by linear hyperplanes. The earliest developments of SVM by Vapnik [CV95] demonstrated the importance of non-linear kernels in certain problem domains.

However, because LibSVM allows for the use of non-linear kernels, it does not contain the same optimizations that LIBLINEAR does for linear kernels, can exhibit much slower training and predicting speed than LIBLINEAR. When predicting, the time to predict new points using LibSVM the time complexity is $O(kn)$ where k is the number of support vectors and n is the number of points to predict, but for LIBLINEAR, the time complexity is $O(n)$. The reason for this is that in LibSVM, to determine the class of a new vector, this vector must first be multiplied by each support vector. This is required in the case of a general non-linear kernel because the dimensionality of the separating hyperplanes in non-linear cases, such as the RBF kernel, can be infinite. For this reason, when using linear kernels, it is preferable to use LIBLINEAR, but for other kernels, LibSVM must be used.

The polynomial kernel is one kernel LibSVM supports and has the form

$$(\gamma * u' * v + coef0)^{degree}$$

In this equation, there are three different parameters that can be varied, the degree of the kernel, γ and $coef0$. To find the optimal performance of this kernel, a parameter search of all these parameters must be carried out. For this experiment, we test different numbers of degrees and different values of gamma and coef0. The results of this parameter sweep are shown in Table 5.12.

The sigmoid kernel is of interest because it is of a similar form to the linear perceptron used in neural networks, a classifier that we will also investigate. The form of the sigmoid kernel is

Degree	C	G	Extract Time	Train Time	Predict Time	Accuracy
1	1.0	1.0	215.08	0.06	0.03	75.00
2	1.0	1.0	214.78	0.06	0.02	76.92
3	1.0	1.0	244.49	0.10	0.02	73.08
10	1.0	1.0	210.70	0.05	0.03	69.23
15	1.0	1.0	209.55	0.05	0.03	65.38
20	1.0	1.0	209.11	0.05	0.03	59.62
3	0.001	0.001	216.25	0.05	0.02	71.15
3	0.001	1.0	232.81	0.06	0.03	69.23
3	0.001	1000.0	300.90	0.11	0.02	73.08
3	0.01	0.001	216.25	0.05	0.02	71.15
3	0.01	1.0	263.05	0.03	0.01	71.15
3	0.01	1000.0	300.17	0.11	0.02	73.08
3	0.1	0.001	241.88	0.05	0.03	71.15
3	0.1	1.0	262.97	0.07	0.03	76.92
3	0.1	1000.0	299.82	0.10	0.02	73.08
3	1.0	0.001	214.71	0.02	0.01	71.15
3	1.0	1.0	244.49	0.10	0.02	73.08
3	1.0	1000.0	299.32	0.13	0.02	73.08
3	10.0	0.001	214.66	0.02	0.01	71.15
3	10.0	1.0	275.85	0.09	0.05	73.08
3	10.0	1000.0	299.32	0.13	0.02	73.08
3	100.0	0.001	214.44	0.02	0.01	71.15
3	100.0	1.0	275.86	0.10	0.02	73.08
3	100.0	1000.0	298.99	0.10	0.03	73.08
3	1000.0	0.001	213.28	0.02	0.01	71.15
3	1000.0	1.0	275.86	0.09	0.02	73.08
3	1000.0	1000.0	298.02	0.10	0.02	73.08
5	1.0	1.0	212.58	0.08	0.03	71.15
7	1.0	1.0	210.70	0.06	0.02	73.08

Table 5.12: Table showing the effects of changing different parameters of the polynomial kernel on the classification performance of LibSVM. The features used in this table include the collection of all audio features described above. In the first half of the table, different degrees of polynomials are tested, and in the second half different values of C and G are tested.

coef0	gamma	Extract Time	Train Time	Predict time	Percent Correct
0.001	0.001	1519.75	57920.86	3847.60	90.18
0.001	1.0	1682.20	49758.06	4544.62	74.26
0.001	1000.0	1780.93	52692.25	4953.43	74.26
0.01	0.001	1519.38	61062.92	3700.08	90.18
0.01	1.0	1682.11	46321.97	4514.10	74.26
0.01	1000.0	1703.92	45367.64	4491.99	74.26
0.1	0.001	1518.33	59801.38	4115.40	90.17
0.1	1.0	1681.37	45547.22	4508.25	74.26
0.1	1000.0	1815.03	51828.98	4977.01	74.26
1.0	0.001	1463.15	49519.48	3756.94	89.98
1.0	1.0	1680.44	47255.89	4706.03	74.26
1.0	1000.0	1801.99	52269.41	4890.40	74.26
10.0	0.001	1511.37	54497.78	5839.89	74.26
10.0	1.0	1679.86	47548.44	4724.58	74.26
10.0	1000.0	1733.04	50049.71	4862.96	74.26
100.0	0.001	1514.19	46585.45	4629.38	74.26
100.0	1.0	1642.95	45031.36	4369.87	74.26
100.0	1000.0	1780.61	50915.99	5137.29	74.26
1000.0	0.001	1520.33	45629.27	4891.16	74.26
1000.0	1.0	1642.95	51583.38	5242.77	74.26
1000.0	1000.0	1709.31	45636.10	4463.55	74.26

Table 5.13: Results showing the effect of changing the values of coef0 and gamma with LibSVM when using a sigmoid kernel. Please note that approximately 3x as many results were calculated but were omitted as they had approximately the same performance as those shown above.

$$tanh(gamma * u' * v + coef0)$$

The parameters for the sigmoid kernel are c and gamma. We did a parameter search of these parameters, the results of which are shown in table 5.13. From this table we can see that the choice of parameters causes wide variations in performance, with the worst performance being 65.77% and the best being 90.18%. This highlights the importance of doing parameter searches for combinations of different datasets and kernels.

From this, we see that the performance was lower than the RBF kernel, but with suitable parameters of C and gamma, we were able to obtain adequate classification performance.

One of the most widely used kernels in the SVM literature [SSB+97] is the Radial

Basis Function (RBF) kernel. The functional form of this kernel is:

$$exp(-\gamma * |\mu - \nu|^2)$$

This kernel, while appearing simple, allows for the creation of hyperplanes of infinite dimension, and works well for many different arrangements of points in space, including ones where a sphere of points of one class is completely surrounded by points of another class. For the RBF kernel, the two parameters are C and gamma. In Table 5.14 results of doing a parameter sweep of these variables is shown. The highest performing set of parameters gave a classification accuracy of 92.15%, which is the best accuracy seen so far, and shows the utility of using different SVM kernels. However, the training time and predicting time for this kernel is very large, with the best accuracy SVM taking 9.4 hours to train and 40.3 minutes to predict. With more data, this training and predicting rate would further decrease and might make it unsuitable when the datasets presented in this thesis are scaled up.

There are many other techniques that have shown their usefulness in the field of Machine Learning besides Support Vector Machines. The field of Machine Learning is under active development, and new techniques are being developed at a rapid rate. In this section, a number of different Machine Learning systems will be evaluated on the orca/background/voice dataset explored in the previous tables.

For this, we will use the Weka [WF05] framework. This framework implements a large number of Machine Learning algorithms. From a search of recent literature, five of the most promising algorithms were chosen, including the Logistic Regression, Decision Trees, Naive Bayes, Random Forest and Multilayer Perceptron algorithms. In the following tables the classification accuracy and time performance of algorithms will be examined.

Logistic Regression [CH92] is a type of linear regression analysis where new instances are classified based on a logistic function.

One of the reasons that this algorithm is useful to consider there exist implementations to distribute it on a large cluster of machines using a map-reduce paradigm [KSRR12] using the Mahout machine learning framework under Hadoop. This could give it the ability to use very large training datasets efficiently.

In the Logistic Regression algorithm, the one parameter that can be adjusted is the value of the ridge in the log-likelihood function R. Results on a parameter sweep of this value is shown in Table 5.15. From this table we can see that this function

Cost	Gamma	Extract Time	Train Time	Predict time	Percent Correct
0.001	0.001	1570.68	92987.46	6851.21	74.26
0.001	0.01	1539.88	89383.63	6368.59	79.41
0.001	0.1	1534.61	81350.14	4826.87	88.10
0.001	1.0	1531.31	120881.52	5763.93	81.75
0.001	10.0	1690.83	89390.93	6611.99	74.26
0.001	100.0	1513.52	204281.47	11600.27	74.26
0.001	1000.0	1479.71	188102.78	11722.92	74.26
0.01	0.001	1570.30	75823.00	6390.49	80.62
0.01	0.01	1537.22	63238.56	4773.81	89.78
0.01	0.1	1533.50	45842.06	3610.45	90.34
0.01	1.0	1529.80	181633.04	6192.49	86.53
0.1	0.001	1570.05	68418.94	4929.80	89.78
0.1	0.01	1536.96	43234.86	4019.78	90.33
0.1	0.1	1533.05	33179.18	2949.36	91.16
0.1	1.0	1524.09	230029.20	5858.43	89.06
1.0	0.001	1569.20	46408.17	3981.22	90.26
1.0	0.01	1532.23	33842.52	3384.35	90.62
1.0	0.1	1533.68	33682.69	2422.72	92.15
10.0	0.001	1665.43	35565.66	3728.54	90.29
10.0	0.01	1535.51	33243.74	2892.56	91.13
10.0	0.1	1533.39	68114.55	1729.90	92.12
100.0	0.001	1535.78	38057.99	3241.33	90.53
100.0	0.01	1557.49	103403.86	2498.79	91.70
100.0	0.1	1531.38	189020.24	1144.29	91.69
1000.0	0.001	1538.36	124636.37	3510.49	90.95

Table 5.14: Table showing the results of using the Radial Basis Function kernel under LibSVM using different values of the Cost and Gamma parameters. Results calculated above use all the audio features described earlier.

Options	Extract Time	Train + Predict time	Percent Correct
-R 1.0	1727.70	3626.55	90.03
-R 1.0E-1	1673.44	3569.68	90.02
-R 1.0E-3	1674.35	3574.83	90.02
-R 1.0E-5	1670.99	3486.96	90.02
-R 1.0E-7	1663.04	3605.77	90.02
-R 1.0E-9	1647.84	3627.77	90.02
-R 10.0	1697.68	2207.96	90.04
-R 100.0	1678.68	1637.44	90.02
-R 1000.0	1685.24	967.19	90.03
-R 10000.0	1688.79	633.62	89.97

Table 5.15: Table showing results of Logistic Regression classifier in the Weka software package with different values of the ridge parameters for the ridge in the log-likelihood function.

gives good performance, with a maximum of 90.04%, and that this accuracy does not vary much when changing the value of R.

This is an exciting result because it means that this algorithm could potentially be used on a large cluster using a Map/Reduce paradigm using the Mahout [OADF11] machine learning framework to speed up training and predicting time. Preliminary results done on a smaller dataset show that this is indeed a practical solution.

Naive Bayes classifiers [JL95] use Bayes theorem for classification. The functional form of Bayes theorem is:

$$P(X|Y) = P(X)P(Y|X)/P(Y)$$

However, the Naive Bayes algorithm assumes that all attributes are independent. This is not the case in many classification problems, but this simple model often works well in practice. In the case of the datasets used in this chapter, the features are in some cases closely interrelated. One advantage of Naive Bayes is that it can be trained efficiently on large datasets.

In the Naive Bayes implementation in Weka, there is one parameter that can be varied, which is if supervised discretization to process numeric attributes should be used. In Table 5.16 the results of using this classifer on the ORCAOBV dataset is shown. The maximum classification accuracy that is obtained is 81.25% which is considerably lower than the other classifiers used, this is likely due to the highly correlated feature vectors that are extracted by the audio feature extraction pipeline.

Decision tree based methods construct a tree in which every node separates the

Options	Extract Time	Train + Predict time	Percent Correct
	1656.21	121.27	75.51
-D	1758.78	380.22	81.25

Table 5.16: Table showing results of Naive Bayes classifier in the Weka package with the -D parameter which corresponds to the use of supervised discretization to process numeric attribute. The results in this table were calculated with a combination of all the previously described audio features.

a set of data into two or more subsets based on the values of different attributes of feature vectors. One such decision tree algorithm is the C4.5 algorithm, which was developed in 1993 by Quinlan [Qui93]. It is similar to the ID3 (Iterative Dichotomiser 3) algorithm [Qui86] and constructs a tree that makes the optimal decision based on information entropy at each leaf

This algorithm starts with all the original data points in one set. At each step it iterates over each leaf of the tree, and looks at the set of instances at that leaf. It goes through all the attributes of that set and finds the attribute that can give the highest information gain (or lowest information entropy) if we were to split the leaf based on that attribute. It then splits that leaf by making a new decision at that leaf splitting the set S into two or more sets using that attribute. This process is iterated over all the leaves until either each element in that leaf has the same label, there are no more attributes to be selected, or there are no more examples in the subset at this leaf.

One advantage of a decision tree based approach is that it is very quick to predict values given a trained classifier.

The J48 algorithm is an open source version of the C4.5 algorithm and is implemented in the Weka framework. In this algorithm, there are a number of different parameters that can be varied, these are described in detail in Table 5.17.

A parameter search over these different parameters was done and results of this are shown in Table 5.17. In this table, the highest classification accuracy was 88.67%, however changing parameters did not have a large impact on classification accuracy or training/prediction time in this case.

The Random Forest model of classification also uses decision trees as described above with the C4.5 classifier, but constructs many different trees using a subset of the features and uses the output of each tree as a vote in the final classification, choosing the label with the most votes. It was originally developed by Breiman and Cutler [Bre01], and uses ideas from bagging, or bootstrap aggregating, where multiple sets of new training sets are generated from an original training set by sampling uniformly from the original training set with replacement.

The algorithm has the advantage that not all the features have to be used in the construction of each new tree like in the C4.5 algorithm, which can give speed benefits, which is somewhat offset by the need to train multiple trees.

It has two parameters that can be adjusted, the number of trees to build "-I" and the number of features to consider "-K". Different values for these parameters

weka	Extract Time	Train + Predict time	Percent Correct
-A	1666.06	2076.96	87.44
-C 0.01	1654.58	2101.03	88.67
-C 0.1	1670.24	2328.97	87.79
-C 0.3	1660.20	2144.50	87.30
-C 0.5	1749.93	2624.74	87.27
-C 0.9	1675.83	2419.89	87.27
-L	1639.50	2277.64	87.44
-M 10	1763.67	2423.18	87.98
-M 3	1893.62	2649.56	87.64
-M 5	1680.84	2194.81	87.73
-N 10 -R	1678.69	1954.46	87.95
-N 3 -R	1687.79	1339.01	87.59
-N 5 -R	1755.38	1983.13	88.07
-R	1676.77	1451.64	87.59
-S	1677.52	2452.46	87.47
-U	1644.99	2131.07	87.29
	1645.00	2240.09	87.44

Table 5.17: Table showing results of J48 decision tree classifier with different values of all adjustable parameters and a combination of all audio features. In this table, the parameter "-A" enables Laplace smoothing for predicted probabilities, "-C" sets the onfidence threshold for pruning, "-L" turns on functionality to not clean up after the tree has been built, "-M" sets the minimum number of instances per leaf, "-N" specifies the number of folds for reduced error pruning, "-R" turns on reduced error pruning, "-S" disables subtree raising, and "-U" uses an unpruned tree.

Options	Extract Time	Train + Predict time	Percent Correct
	1665.07	325.55	91.30
-I 1	1732.44	86.57	85.37
-I 2	1659.33	112.11	83.97
-I 3	1628.72	142.09	89.02
-I 4	1621.68	163.20	88.93
-I 5	1608.67	188.73	90.20
-I 7	1606.52	246.34	90.76
-I 10	1619.33	326.33	91.30
-I 20	1594.33	547.85	91.62
-I 50	1577.71	1360.71	91.97
-K 1	1656.54	106.36	88.90
-K 2	1667.73	133.08	90.03
-K 3	1748.82	205.72	90.32
-K 4	1641.89	193.03	90.59
-K 5	1666.93	222.82	90.96
-K 7	1666.43	296.44	90.97
-K 10	1666.43	370.04	91.22
-K 20	1686.20	685.11	91.18
-K 50	1646.31	1400.37	91.44

Table 5.18: Table showing results of random forest classifier with different values of the number of trees to build (-I) and the number of features to consider (-K). Like in previous tables, all the audio features described in this chapter was used as input to the random forest classifier.

were tested and the results are shown in Table 5.18. From this table we can see that the performance of the Random Forest classifier was quite good, with a maximum performance of 91.97%. In addition, the training and testing time for this algorithm was excellent, with the highest accuracy having a training + predicting time of 5.4 minutes.

This algorithm is also amenable to large scale parallelization of the training step, and is potentially a good algorithm to use when larger datasets are constructed out of the data in the Orchive.

A Multilayer Perceptron [RRK+90] is a neural network classifier that uses back-propogation to train the algorithm. All nodes use a sigmoid function. It uses a similar to Deep Belief Networks (DBN) [RHW02], which are one of the most promising set of machine learning when exploring large, high dimensional datasets such as handwriting recognition. However DBNs can be trained more efficiently because of the use of a pre-training step using Restricted Boltzman Machines (RBM) [HOT06]

Options	Extract Time	Train + Predict time	Percent Correct
	1687.34	65735.35	88.25
-H 0 -N 5	1693.06	187.28	87.93
-H 1 -N 5	1676.01	173.37	85.50
-H 10 -N 5	1666.87	227.41	88.85
-H 100 -N 5	1749.07	1072.38	88.59
-H 2 -N 5	1682.23	181.21	87.82
-H 20 -N 5	1668.37	305.75	89.97
-H 5 -N 5	1666.24	190.82	89.37
-H 50 -N 5	1733.19	585.73	88.34
-L 0.01	1638.87	65838.11	90.66
-L 0.1	1655.32	65136.71	91.04
-L 0.2	1652.85	64794.65	90.26
-L 0.5	1655.64	66352.95	89.89
-L 0.7	1662.33	67643.16	90.51
-L 0.9	1647.23	67100.37	89.91
-M 0.01	1723.80	65911.87	90.17
-M 0.1	1639.76	65624.59	90.66
-M 0.2	1655.10	65363.26	88.25
-M 0.5	1655.11	57785.01	89.57
-M 0.7	1635.97	137770.83	76.00
-M 0.9	1665.08	73378.21	86.98
-N 10	1647.39	1533.55	88.79
-N 1	1643.36	381.99	90.41
-N 5	1682.43	876.23	89.86

Table 5.19: Table showing results of multilayer perceptron classifier with different values of a variety of parameters, including the number of hidden layers and parameters for the backpropogation algorithm.

The Multilayer Perceptron algorithm as implemented in Weka has a number of parameters that can be set. The first is the number of hidden layers H, which is the number of layers of neurons between the input and output layers. Other parameters are the number of epochs to train through "N" and the learning rate "L" and the momentum rate for the backpropogation algorithm.

The results of the MultilayerPerceptron algorithm are shown in Table 5.19. In this table, one can see that the maximum peformance obtained is 91.04% which is competitive with the other machine learning algorithms tested.

Because of this, it could be useful to test the more efficiently trainable Deep Belief Networks on this dataset.

5.2 Call classification

Once the large amount of data has been segmented into sections that have been identified as orca calls, we are now interested in classifying these calls based on the call catalog of Ford [For87]. As an additional and related task, we are also interested in classifying these calls based on the clan, pod, subpod and matriline that made the calls. These are overlapping tasks as some calls are only made by one pod so classifying some of them would automatically give the pod they came from, like the N47 call. Other calls are very distinct from pod to pod, and can be handled as separate classification tasks, and others are so similar, like N3, that it is difficult for an untrained person to differentiate between calls made by different pods, and would likely be a difficult machine learning task. We call the call catalog dataset used in this section ORCACALL. It contains 2954 different clips containing orca calls as labelled by people trained to do Northern Resident Killer Whale (NRKW) orca call classification.

It should be noted though that there are other forms of information that can be used by scientists in the future when using classifiers from this work, the most important being direct visual observation combined with the use of the photo identification catalog. This could be combined with acoustic arrays of hydrophones to localize individual whales.

In this section, we will follow a similar strategy to that for the classification task above, and will discuss the similarities and differences when classifying calls as opposed to simply classifying audio into orca/background/voice as in the ORCAOBV dataset. The dataset with calls will be referred to as ORCACALL in the following tables.

In order to classify the clips in the ORCACALL dataset, a different experimental procedure to that used in the ORCAOBV results above. Each clip was considered to have been presegmented by either the user or the machine learning system and results on a per clip level were generated. These clip level decisions were made using a voting metaphor was used, where each feature vector in a clip was assigned a label by the classifier individually, and the clip was labeled with the label that occurred the most often.

Following a similar procedure to that carried out for ORCAOBV dataset, a parameter scan of different window sizes, hop sizes and texture window sizes was performed, with results shown in Table 5.20. In this table there is a larger variation in accuracy

ws	hs	win	Extract Time	Train Time	Percent Correct
512	256	1	141.79	87.32	47
512	256	2	127.92	505.95	47
512	256	5	153.62	314.00	48
512	256	10	140.79	339.46	48
512	256	20	118.44	368.40	49
1024	512	1	90.56	30.82	44
1024	512	2	117.79	177.41	48
1024	512	5	108.07	178.74	48
1024	512	10	102.28	137.23	50
1024	512	20	92.13	151.60	51
2048	1024	1	86.24	13.75	44
2048	1024	2	83.15	118.59	49
2048	1024	5	72.37	69.04	49
2048	1024	10	85.48	62.31	52
2048	1024	20	78.48	85.45	52
4096	2048	1	76.07	5.34	46
4096	2048	2	83.15	38.11	49
4096	2048	5	72.37	30.38	51
4096	2048	10	75.06	28.24	52
4096	2048	20	77.32	27.35	52

Table 5.20: Table of MFCC results with different window sizes using the LIBLINEAR classifier. In this and subsequent tables, "ws" refers to the size of the FFT window in samples, "hp" refers the hop size between sequent FFT frames in samples, and "mem" refers to the size of the texture window in frames. Longer texture window sizes have been omitted as they showed anomalous behaviour due to the short size of the clips as compared to the long integration time.

than was seen in the case of the ORCAOBV dataset, and that the highest accuracy that was obtained was 52%, which was obtained in several experimetntal conditions. Because of this similar performance for subsequent tables, a window size of 2048, hop size of 1024 and window size of 10 was used to facilitate comparison with the results for the ORCAOBV dataset.

As in the ORCAOBV dataset, a parameter scan of the number of MFCC components was carried out, with results shown in Table 5.21. The highest performance obtained was for a window size of 4096 and 100 MFCCs with an accuracy of 58%, with a window size of 2048 and 100 MFCCs yielding an accuracy of 54%. These results show a similar trend to the results obtained for ORCAOBV with more MFCCs giving better results. Because of the longer size of window that results from using

a bigger window, a window size of 2048 and 100 MFCCs will be used in subsequent tables.

A parameter sweep of using different solvers and values of C for LIBLINEAR was then carried out, with results shown in Table 5.22. In this table, several experimental conditions gave the same performance of 54%, however unlike in the ORCAOBV case, there was considerably more variation in performance with this classifer. In terms of speed, the L2-regularized L2-loss support vector classification (primal) gave the fastest consistent performance.

In order to investigate the manner in which different audio features affect classification performance, a series of experiments was carried out in which different audio features and combinations of these audio features was used, with a similar methodology to that used on the ORCAOBV dataset.

In the first test, different window and hop sizes were used as input to the MFCC algorithm, with a texture memory size set at 10, these results are shown in Table 5.23. Unlike in the ORCAOBV case, the best classification accuracy was obtained for longer window sizes, with the best performance being at a window size of 4096 which gave a classification accuracy of 53%. When the RMS feature alone was used, a dramatic drop in performance was seen, with the accuracy in each window size being 43%. This is an expected behaviour because with the ORCACALL database the loudness of different calls is uncorrelated with the call type, and is dependant instead on the distance of the orca from the hydrophone.

Surprisingly, the use of statistical measures of the spectrum, as shown in Table 5.25 gave poor classification accuracy, with a maximum classification accuracy of 43%. This is likely because the broad spectral characteristics of these measures did not accurately capture important factors about the differences between calls. However, when using a combination of Chroma, SFM and SCF as shown in Table 5.26, a classification accuracy of 59% was obtained, which was higher than using MFCCs alone, but unlike the case for ORCAOBV, this higher classification accuracy was not universally seen, but was only observed in one experimental condition.

When using the YIN pitch determination algorithm as is shown in Table 5.27 results were poor, with a maximum accuracy of 43%. This is a surprising result, as one would hope that the YIN algorithm would capture the pitch contour of the call and would therefore lead to good classification. Upon examination of the data with the Orchive Version 2.0 interface, the cause of this appeared to be because the low signal/noise ratio (SNR) of the call. If better pitch estimation could be carried out,

ws	hp	Num MFCCs	Train Time	Percent Correct
512	256	1	30.86	43
512	256	5	125.01	43
512	256	10	217.63	48
512	256	15	325.26	49
512	256	25	675.81	51
512	256	30	1059.44	51
512	256	40	1153.37	51
1024	512	1	15.20	43
1024	512	5	50.34	43
1024	512	10	122.80	49
1024	512	15	155.82	50
1024	512	25	336.11	52
1024	512	50	836.08	53
1024	512	70	1850.25	53
1024	512	100	2667.66	53
2048	1024	1	5.24	43
2048	1024	5	22.52	44
2048	1024	10	50.33	49
2048	1024	15	75.94	51
2048	1024	25	192.55	53
2048	1024	50	524.04	54
2048	1024	70	887.34	54
2048	1024	100	1474.60	54
4096	2048	1	2.51	43
4096	2048	5	9.18	44
4096	2048	10	21.19	50
4096	2048	15	34.65	54
4096	2048	25	78.55	57
4096	2048	50	213.46	58
4096	2048	70	346.59	58
4096	2048	100	668.68	58

Table 5.21: Table of MFCC results with different numbers of MFCC coefficients using the LIBLINEAR classifier and the ORCACALL dataset

s	c	Train Time	Percent Correct
1	0.001	3.71	51
1	1.0	31.16	53
1	100.0	405.72	43
2	0.001	3.55	51
2	1.0	3.73	53
2	1000.0	3.78	53
3	0.001	3.40	51
3	1.0	18.84	54
3	1000.0	425.39	33
4	0.001	2.28	49
4	1.0	32.90	52
4	1000.0	9557.33	46
5	0.001	15.43	47
5	1.0	34.04	53
5	1000.0	31.91	53
6	0.001	5.52	43
6	1.0	14.32	54
6	1000.0	17.12	54
7	0.001	5.20	45
7	1.0	12.45	54
7	1000.0	635.40	50

Table 5.22: A table showing the results with the ORCACALL dataset of doing a parameter search over a wide number of values of C and using the different solvers in the LIBLINEAR package.

ws	hp	Extract Time	Train Time	Percent Correct
256	128 -m 10	228.01	70.64	47
512	256 -m 10	176.47	28.56	48
1024	512 -m 10	131.70	15.56	49
2048	1024 -m 10	102.47	7.60	51
4096	2048 -m 10	98.47	3.62	53

Table 5.23: Table showing the classification accuracy and timing when using different FFT window sizes on MFCC features using a linear SVM kernel using the ORCACALL dataset.

Options	Extract Time	Train Time	Percent Correct	
1024	512 -m 10	100.13	1.66	43
2048	1024 -m 10	88.20	0.82	43
256	128 -m 10	178.01	7.00	43
4096	2048 -m 10	86.54	0.33	43
512	256 -m 10	135.04	3.41	43

Table 5.24: Table using audio from the ORCACALL dataset showing the effect of window size on classification performance with LIBLINEAR using the Root Mean Square (RMS) energy of a signal as a feature.

the classification accuracy would likely be higher when using this feature.

When all these audio features were combined however, a large increase in performance to a maximum of 62% was achieved, as can be seen in Table 5.28. This value was seen in three separate experimental conditions, and unlike in the case of ORCAOBV where chroma+SFM+SCF features alone gave approximately the same classification accuracy as all features, in the ORCACALL dataset, using all the features was considerably better than using just the chroma+SFM+SCF features. This is likely due to the increased complexity of the classification task when doing call classification, and that the extra features captured different information about the call.

Experiments using LibSVM using the sigmoid kernel showed a decrease in classification performance over using a linear kernel, with a maximum classification accuracy of 54% in one experimental condition as is shown in Table 5.29. This is surprising because of the more complex form of the sigmoid kernel as compared to the linear kernel, and might be due to the fact that the features used could not be well modelled by a sigmoid kernel. On the other hand, when using the RBF kernel, excellent classification results were obtained as is shown in Table 5.30, with a maximum classification accuracy of 74%. This is an expected result and is seen in many cases when using

Feature	ws	hp	Extract Time	Train Time	Percent Correct
centroid	1024	512 -m 10	119.32	1.58	43
centroid	2048	1024 -m 10	102.19	0.85	43
centroid	256	128 -m 10	215.26	6.99	43
centroid	4096	2048 -m 10	96.78	0.32	43
centroid	512	256 -m 10	144.23	3.10	43
flux	1024	512 -m 10	130.55	1.69	43
flux	2048	1024 -m 10	107.72	0.91	43
flux	256	128 -m 10	218.31	7.43	43
flux	4096	2048 -m 10	95.93	0.39	43
flux	512	256 -m 10	145.60	3.49	43
rolloff	1024	512 -m 10	117.73	1.42	43
rolloff	2048	1024 -m 10	109.10	0.71	43
rolloff	256	128 -m 10	205.94	6.08	43
rolloff	4096	2048 -m 10	96.30	0.30	43
rolloff	512	256 -m 10	151.70	6.37	43
kurtosis	1024	512 -m 10	123.56	1.91	43
kurtosis	2048	1024 -m 10	108.91	0.74	42
kurtosis	256	128 -m 10	212.50	7.14	42
kurtosis	4096	2048 -m 10	95.63	0.39	42
kurtosis	512	256 -m 10	145.55	3.30	42
skewness	1024	512 -m 10	202.36	1.79	43
skewness	2048	1024 -m 10	317.77	0.66	42
skewness	256	128 -m 10	232.54	7.31	43
skewness	4096	2048 -m 10	571.55	0.38	43
skewness	512	256 -m 10	179.31	3.51	43

Table 5.25: Table showing the effect of using combinations of different statistical measures of the spectrum of a signal on classification performance with LIBLINEAR.

Features	ws	hp	Extract Time	Train Time	Percent Correct
chroma	1024	512	134.10	14.60	43
chroma	2048	1024	107.68	7.53	43
chroma	256	128	271.09	62.35	43
chroma	4096	2048	89.70	4.42	43
chroma	512	256	188.57	28.86	43
scf	1024	512	131.62	35.30	43
scf	2048	1024	107.43	17.67	45
scf	256	128	286.35	148.22	43
scf	4096	2048	99.94	11.43	51
scf	512	256	192.18	70.59	43
sfm	1024	512	152.44	30.12	46
sfm	2048	1024	114.02	16.99	47
sfm	256	128	311.21	167.47	43
sfm	4096	2048	100.68	8.82	52
sfm	512	256	203.01	71.52	43
all	1024	512	218.39	163.69	48
all	2048	1024	147.61	74.98	53
all	4096	2048	124.08	39.91	59
all	512	256	321.27	308.60	45

Table 5.26: Table showing the classification performance of LIBLINEAR with different combinations of Chroma, Spectral Crest Factor (SCF) and Spectral Flatness Measure (SFM).

ws	hp	Extract Time	Train Time	Percent Correct
256	128	181.64	6.17	43
512	256	166.88	2.16	43
1024	512	213.81	1.31	43
2048	1024	337.15	0.74	42
4096	2048	560.11	0.41	43

Table 5.27: Table showing the result of using the YIN pitch estimator as a feature for input to the LIBLINEAR SVM classifier with different window sizes and hop sizes.

Options	ws	hp	Extract Time	Train Time	Percent Correct
spectral sfm scf	1024	512	346.72	155.96	54
spectral sfm scf	2048	1024	422.53	92.64	56
spectral sfm scf	4096	2048	632.05	42.52	62
spectral sfm scf	512	256	399.36	291.94	54
spectral css	1024	512	356.51	189.12	55
spectral css	2048	1024	432.85	101.64	56
spectral css	4096	2048	648.42	48.30	62
spectral css	512	256	420.97	335.38	54
spectral css yin	1024	512	495.21	192.56	55
spectral css yin	2048	1024	663.14	96.12	57
spectral css yin	4096	2048	1105.28	45.16	61
spectral css yin	512	256	519.07	362.27	54
spectral css yin rms	1024	512	489.94	197.62	55
spectral css yin rms	2048	1024	665.80	104.32	57
spectral css yin rms	4096	2048	1134.22	53.16	62
spectral css yin rms	512	256	534.51	358.81	54

Table 5.28: Table showing the use of all audio features described above as input to the LIBLINEAR package. In this table "css" refers to the combination of Chroma, SCF and SFM features.

r	g	Train Time	Percent Correct
0.001	0.001	55574.30	45
0.001	1.0	20541.20	43
0.001	1000.0	16521.20	43
0.01	0.001	51793.95	45
0.01	1.0	16313.86	43
0.01	1000.0	17835.71	43
0.1	0.001	38669.18	45
0.1	1.0	19151.06	43
0.1	1000.0	18491.59	43
1.0	0.001	39542.84	44
1.0	1.0	18575.09	43
1.0	1000.0	16522.39	43
10.0	0.001	21918.30	43
10.0	1.0	16359.08	43
10.0	1000.0	18169.31	43
100.0	0.001	18785.54	43
100.0	1.0	19176.62	43
100.0	1000.0	16686.64	43
1000.0	0.001	20761.40	43
1000.0	1.0	16334.70	43
1000.0	1000.0	20675.97	43

Table 5.29: Results showing the effect of changing the values of coef0 and gamma with LibSVM when using a sigmoid kernel on the ORCACALL dataset.

the RBF kernel on complex datasets in which feature vectors are not easily linearly separable. The increase in performance of 11% over a linear kernel was considerably higher than the ORCAOBV case where the increase in classification accuracy was only 2.58%. This is a dramatic result, and shows the importance of using more complex kernels such as RBF on more complex datasets.

The Logistic Regression classifier on the ORCACALL dataset gave relatively poor performance, with a maximum classification accuracy of 63%. This is dissimilar to the performance of the Logistic Regression algorithm in the ORCAOBV dataset where good classification accuracy was obtained, and is likely due to the more complex distribution of feature vectors in this dataset. However, this performance was much better than the Naive Bayes algorithm as seen in Table 5.32 which gave a maximum classification accuracy of only 31%, which is considerably higher than random, but is relatively much lower than its performance on the ORCAOBV dataset.

The J48 classifier gave good classification accuracy results, with a maximum ac-

c	g	Train Time	Percent Correct
0.001	0.001	43584.46	43
0.001	1.0	40204.36	43
0.001	1000.0	68547.56	43
0.01	0.001	37131.36	43
0.01	1.0	68410.96	43
0.01	1000.0	175016.63	43
0.1	0.001	37727.36	43
0.1	1.0	94218.22	49
1.0	0.01	25538.28	57
1.0	0.1	22124.17	69
10.0	0.01	25344.99	65
100.0	0.001	28406.91	59
100.0	0.01	57523.64	69
1000.0	0.001	67110.49	65
1000.0	1.0	163658.65	60

Table 5.30: Table showing the results of using the Radial Basis Function kernel under LibSVM using different values of the Cost and Gamma parameters.

R	Train Time	Percent Correct
1.0	102558.01	63
10.0	64488.55	63
100.0	37770.69	61
1000.0	25822.61	57
10000.0	9779.39	50

Table 5.31: Table showing results of Logistic Regression classifier in the Weka software package with different values of the ridge parameters for the ridge in the log-likelihood function.

Options	Train Time	Percent Correct
-D	1430.86	31
-K	292.00	14
	308.50	09

Table 5.32: Table showing results of Naive Bayes classifier in the Weka package with the -D parameter which corresponds to the use of supervised discretization to process numeric attribute.

Options	Train Time	Percent Correct
	11197.11	71
-A	11235.84	71
-C 0.01	10877.68	72
-C 0.1	9768.69	71
-C 0.3	10193.41	71
-C 0.5	9939.26	71
-C 0.9	9766.73	71
-L	10674.71	71
-M 10	9667.26	74
-M 3	10705.30	69
-M 5	9916.05	71
-N 10 -R	9389.09	71
-N 3 -R	6466.16	68
-N 5 -R	8816.36	71
-R	6956.54	68
-S	11454.72	71
-U	9951.19	71

Table 5.33: Table showing results of J48 decision tree classifier with different values of all adjustable parameters and a combination of all audio features. In this table, the parameter "-A" enables Laplace smoothing for predicted probabilities, "-C" sets the onfidence threshold for pruning, "-L" turns on functionality to not clean up after the tree has been built, "-M" sets the minimum number of instances per leaf, "-N" specifies the number of folds for reduced error pruning, "-R" turns on reduced error pruning, "-S" disables subtree raising, and "-U" uses an unpruned tree.

curacy of 72%, only 1% lower than SVM with an RBF kernel. Furthermore, several other experimental conditions gave a 71% classification accuracy accuracy. This is surprising given the straightforward nature of the J48 algorithm, and is would be an interesting area for future research. Of note is that the Random Forest algorithm only gave a 1% improvement to this result as is shown in Table 5.34. This seems to be related to the fact that many different experimental conditions of the J48 algorithm gave similar performance, and seems to indicate that the C4.5 algorithm is capable of easily finding structure in the feature vectors for this dataset.

Surprisingly, the Multilayer Perceptron gave poor performance, with a maximum classification accuracy of 58%, as is shown in Table 5.35. However, this is likely due to the fact that many of the experimental conditions that were tried were unable to be run in less than 3 days, which is the maximum wall clock time available on the Hermes/Westgrid cluster. This is due to the inefficient way that the backpropogation

Options	Train Time	Percent Correct
	1628.82	70
-I 10	1507.55	70
-I 1	192.74	69
-I 20	3148.37	69
-I 2	324.72	67
-I 3	490.17	70
-I 4	627.04	70
-I 50	7550.82	70
-I 5	796.59	70
-I 7	1278.49	71
-K 10	1925.79	71
-K 1	320.77	58
-K 20	4002.33	72
-K 2	522.78	64
-K 3	671.68	68
-K 4	864.01	68
-K 50	9493.45	73
-K 5	1018.99	69
-K 7	1374.28	70

Table 5.34: Table showing results of random forest classifier with different values of the number of trees to build (-I) and the number of features to consider (-K). Like in previous tables, all the audio features described in this chapter was used as input to the random forest classifier.

Options	Train Time	Percent Correct
-H 0 -N 5	768.62	58
-H 1 -N 5	353.24	43
-H 10 -N 5	695.42	54
-H 100 -N 5	4271.41	25
-H 2 -N 5	366.75	46
-H 20 -N 5	1008.34	58
-H 5 -N 5	464.23	54
-H 50 -N 5	2261.97	44
-N 10	7360.12	31
-N 1	1004.90	56
-N 5	3807.54	20

Table 5.35: Table showing results of multilayer perceptron classifier with different values of a variety of parameters, including the number of hidden layers and parameters for the backpropogation algorithm.

algorithm is implemented in Weka, a factor that is specifically addressed by Deep Belief Networks.

Considerable work was undertaken to run these results using Theano [BBB+10] Deep Belief Network package. Theano requires considerable time and expertise to setup in its most efficient GPU configuration, and using the available hardware and was unfortunately unable to be used. However, work is currently underway to obtain these results using a Deep Belief Network system such as Theano. This could potentially increase the maximum classification accuracy of 74% that was obtained in this section.

The purpose of these experiments is to find the combination of audio features and machine learning algorithm that gives the highest performance, and to then use this combination to predict classes for all recordings in the Orchive. In order to investigate the performance of the classification of recordings into Orca, Background and Voice, we trained a SVM with a section of 30 and 240 seconds of hand trimmed data using the bextract program in Marsyas. We then used the sfplugin program in Marsyas to classify all the recordings in the Orchive on the Hermes/Nestor cluster, part of the Westgrid computational resource. For this we divided the data into sets of 1%, 5%, 10% and 100% of the Orchive. The timing results of these datasets run on 10 computers are shown in Table 5.36. From this we can see that the classifier that had more data took longer to classify, and that the speedup from taking samples of the data was almost linear.

Training data (sec)	% of Orchive	Run time (d:h:m:s)
30	1	00:00:05:18
30	5	00:00:25:20
30	10	00:00:50:58
30	100	00:09:01:05
240	1	00:06:16
240	5	00:00:31:21
240	10	00:04:47:12
240	100	02:04:18:32

Table 5.36: Performance results of timing on subsets of the entire Orchive dataset.

These results were generated using the bextract/sfplugin combination from Marsyas, which allows for feature extraction and machine learning in a single executable, which allows for a dramatic speedup because audio features do not need to be written to disk and reread from disk. In this hybrid program, audio features train the machine learning system directly. At the current time, Marsyas only supports LibSVM which limits its usefulness as compared to LIBLINEAR on huge training datasets. With LibSVM the predicting time is linear in the number of training samples multiplied by the predicting samples, but with LIBLINEAR, the predicting time is only linear in terms of the number of features to be predicted. With the approximately 6000 seconds of audio data in the ORCAOBV dataset, the time to classify all the audio in the Orchive would take 10 days on 50 computers, and would take a fraction of that time using LIBLINEAR. For this reason, work is underway on adding LIBLINEAR to Marsyas.

5.3 Symbolic Approximation of Pitch Contours

The transcription of time series data into a character sequence that can be consumed by bioinformatics sequence alignment tools is a nontrivial task. An attempt towards this goal is Symbolic Aggregate Approximation (SAX) which was proposed by Lin et al. in [LKLC03]. The underlying idea of SAX is to parse a time series using a sliding window and to generate a character sequence that approximates the signal's normalized slope using a technique called Piecewise Aggregate Approximation (PAA). SAX is most useful in cases where the data is not on an absolute scale. In order to investigate the utility of SAX on this dataset, we plotted the fundamental frequency

curves of a number of examples of calls to each other, two of these plots are shown in Figure 5.2 and Figure 5.3. From these we can see that the absolute pitch of these calls is well conserved, a result that has been previously observed [For87]. It is of interest to note that the N05 call voiced by the A35 matriline is of a lower pitch than the others, this matriline is in the A4 pod, while the calls by the A12 and A36 matrilines are of more similar pitch to one another. This information could be used to help classify which pod is vocalizing a particular call.

Figure 5.2: F0 contour for 21 examples of the N04 call.

Figure 5.3: F0 Contour for 10 examples of N05 call.

This approach is certainly valid if slope changes within a given window characterize

the signal well and the signal is fairly free of noise. We therefore have examined SAX for a possible avenue to transcribe Orca vocalizations. We determined, however, that considering windowed slope fragments of Orca vocalizations does produce sequences that are of lower quality than those produced by our transcription technique which we refer to as FTSQ (Fundamental Frequency Time Series Quantization). The FTSQ procedure is outlined in detail below. We do not provide classification results for SAX sequences because the majority of SAX sequences were too long in order to be processed by our dynamic programming based alignment algorithm and hence a direct comparison can't be impartial. Further, we observed that the fundamental frequency of Orca vocalizations carries a lot of information, that is, it seems to be stable, and within a certain range that is unique to a large portion of the vocalizations in our call catalog. Since SAX is normalizing the signal slope within a sliding window it cannot take advantage of this important feature.

Fundamental Frequency Time Series Quantization is a time series quantization approach that transcribes an audio time series based on an estimate of its fundamental frequency (F_0).

In general it is not feasible to analyze Orca vocalizations as a raw time series, since these signals are a complex mixture of sinusoids and noise (see Figure 5.4). Figure 5.5 shows a spectrogram representation of the N01 and N09 Orca vocalizations. When comparing Figures 5.4 and 5.5 it is easy to see that the frequency components of the audio signal reveals much more about the structure of the signal than a raw audio time series by itself.

Figure 5.4: Raw audio signal of Orca vocalization N01.

Quantizing the fundamental frequency of an audio signal is am approach that makes use of the underlying structure in the frequency bands of the audio signal. Loosely speaking one may think of the fundamental frequency as the pitch of a sound signal. For our FTSQ technique we use Yin [dCK02] a robust fundamental frequency

Figure 5.5: Frequency spectrum of Orca vocalizations N01 and N09.

estimation algorithm developed by Cheveigne et al. Processing the audio waveform from Figure 5.4 with Yin yields the fundamental frequency (in octaves relative to 440 Hz) shown in the top plot of Figure 5.6. The two plots below the fundamental frequency show the aperiodicity and the period-smoothed instantaneous power of the signal; they give estimate in the confidence on the approximation of F_0 and are used to identify meaningful regions of interest within the F_0 signal.

A further issue that is encountered commonly in the process of determining a pitch contour for an audio recording is the optimization of all the parameters of the pitch detection algorithm to perform well on the particular dataset that is being used. There are many such parameters for each different algorithm, the important ones in the Yin algorithm include the window size and hop size of the FFT, the high and low frequency cutoffs, the high and low frequencies around which the pitches will be wrapped using modulo arithmetic, the tolerance of the Yin algorithm which determines which peaks in the autocorrelation will be used for the pitch determination, the window size for the median filter, and the number of histogram bins in which to divide the frequency range into. In our previous work, this was done by hand by plotting points using MATLAB or Gnuplot, which can be a long and labour intensive process.

For this project, we added custom visualization tools to our OpenMIR platform to view the original audio as a spectrogram, to allow the user to listen to this audio, to show a pitch contour with dynamic controls over these various parameters, and an energy display that shows the RMS energy of the audio signal. This interface is shown in Figure 5.7. This software allowed us to quickly iterate over a large number of parameters, and to determine the optimal parameters for the Yin algorithm. The

Figure 5.6: Frequency spectrum of Orca vocalizations N01 and N09

most important of these was the median filter size, and a median filter of size 11 is shown in Figure 5.7.

The output of Yin does not always give a proper estimate of F_0. Indeed, the F_0 signal in Figure 5.6 displays several estimation errors that we herein refer to as *octave errors*. Octave errors manifest themselves as sudden jumps in a multiple of the core F_0 frequency. Figure 5.6 displays multiple octave errors the first one at about 0.1 seconds. When quantizing the F_0 signal into a letter representation over a finite alphabet. Octave errors result in mis-mapped letters in the output sequence. To a certain degree we are able to handle octave errors by crafting a custom substitution matrix in our sequence alignment algorithm. However, doing so ultimately lowers the confidence in our alignment score and might confuse legitimate frequency jumps with octave errors and as a result would not penalize mismatches appropriately. Therefore, we have developed a technique that can automatically correct for the majority of

Figure 5.7: Spectrogram, pitch and RMS view of a small portion of the Orchive catalog viewed with the OpenMIR interface.

octave errors and as a net effect produces a more accurate letter representation.

We first run the F_0 signal obtained from Yin through a median filter, which smoothes the signal and eliminates small transients that occur due to noise in the original signal. We then threshold the aperiodicity and the period-smoothed instantaneous power to obtain F_0 signal regions at which the F_0 signal is estimated with high confidence. After down-sampling the signal we obtain the representation shown in Figure 5.8.

The green line represents the median of the original F_0 signal drawn in blue. The octave errors are clearly identifiable as such in Figure 5.9 which shows versions of the F_0 signal one octave above (purple) and on octave below (red) the original (blue). F_0 signal. To correct for octave errors we simply piece the portions of the three versions of the F_0 signal together according to which ever signal is closest to a slightly upwards biased median of the original signal (blue). This yields the black F_0 curve wich shows the now automatically corrected version of the F_0 signal.

When transcribing the F_0 signal into a letter representation suitable for sequence alignment tools it is natural to ask how these letters should be assigned over the range of possible F_0 frequency values. A naive approach would be to map F_0 frequencies to letters in a linear fashion. However, while this approach does work well in practice,

Figure 5.8: Thresholded and down-sampled F_0 signal of a N01 call.

we investigated if a non-linear mapping might potentially be more appropriate. For this purpose we plotted the distribution of F_0 frequencies over time for the whole catalog of Orca vocalizations in Figure 5.10. The dynamic range of F_0 incorporates frequencies from about 80 to 2400Hz, with three high density bands at around 250, 700 and 1200Hz. Ideally one would therefore quantize the F_0 signal using a three-modal distribution. However, for simplicity we chose a log-normal distribution (see Figure 5.11), which provides us with a fine resolution for the more common F_0 frequencies and quantizes high F_0 frequencies at a coarser level.

The result of log-normal quantization of the F_0 trace in Figure 5.9 is provided in Figure 5.12. Finally, the resulting letter sequence is given by:

`LLHHJJKKKKKKKKKKKKJJJJJJJII--IIIIIHHHHHHHGGGGGFFFFFF`

which can readily be consumed by a dynamic programming sequence alignment algorithm.

In order for sequence comparison algorithms to work well, it is necessary that the combination of the letters and the scoring matrix are compatible and output meaningful results. In the two sections below, we show the result of quantizing the signal

Data Set	Global Accuracy
Original	64%
Octave Removal - Linear	81%
Octave Removal - NonLinear 900	80%
Octave Removal - NonLinear 1200	81%
J48	58%
Naive Bayes	65%
SVM	75%

Table 5.37: Global accuracy for different methods of converting a Yin pitch contour into an alphabet using FTSQ

Call	Accuracy
N47	0.400
N01	0.969
N12	0.583
N05	0.785
N04	1.000
N03	0.75
N09	0.772

Table 5.38: Global accuracy for different call types using Octave Removal NonLinear 1200 parameters with the FTSQ algorithm.

Figure 5.9: Octave errors within a F_0 signal of a N01 call.

using the nonlinear histogram approach for two of the more common calls vocalized by A-clan Northern Resident whales. The first is the very common N4 call, which consists of a quick up swing in pitch, followed by a downswing and then a constant tone. In it we can clearly see long regions of the repeated letters, P,O,N,M,L,K. Even from a quick visual inspection these repeated letters show that our method of converting frequencies into a discrete alphabet is promising.

```
N04 A04 HIKNOOOOOPPPPOOOOOOOOOOOONNNNNNNNNNNNMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKKKKKK
N04 A04 IJNOPPPPPPPPPPPOOOOOOOOOONNNNNNNNNMMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKKKKKK
N04 A05 NLKLMNOOOOOOOOOOOOOOOOOOOOOOOOOOOONNNNNNNNNNNNNNNNNNNNMHGQ
N04 A08 RILMNNOOOOOOOOIIOOOOOOOOOOONNNNNNNNNNNNNNNNNNNNMMMMMMMMMMMMMMMMMMMMLLLLLLLMLLLLL
N04 A08 SIIJJKNPQQRPIIOOOOOOOONNNNNNNNNNNNNNMMMMMMMMMMMMMMMMMMMMMMMMMMMDCCCMMM
```

Even more promising were the results for another call, N5, which consists of a constant tone. These calls are from the A36 matriline of orcas, which now consists of only two whales, the brothers A37 and A46, although the grandmother whale, A12 often now associates with this matriline after A34, her daughter's matriline, grew large in size. From these calls, we can see that the frequency represented by the letter L is very constant throughout the entire call, and is a clear indication that this is an N5 call.

Figure 5.10: Distribution of F_0 frequencies of call catalog.

```
N05,A36,A36-N05-070806-D012-13913,IIJJJKKKKKLLL
```

Figure 5.11: Log-normal distribution for quantization

play more rounds of the game. The full game interface is shown in Figure 5.21.

This game was designed to be the simplest possible game like interface that would gather data for this project. Last year, we had written a much more involved interface that had many more features, such as multiple turns per level where the user was first presented with only the sounds and was asked to guess the clip, then if they got it wrong they were allowed to see the spectrogram for a brief period of time and guess again, and finally were allowed to see the spectrogram. This interface also had a running score counter, where the user would get points for correct answers, and if they answered correctly with fewer hints in a level, they were rewarded with a higher score. This interface also had "achievements" which were an in game reward given for correctly identifying 10 calls of a specific type, for example the "N01" call. Another form of reward given was that after a series of levels were completed, the user was given a screen showing them some information about orcas along with pictures of orcas. We hope to add these additional features back into the game at some point, but were interested in the current work if the simplest possible game mechanic was engaging for users.

For this study, we chose the clips used in the game by hand in order to evaluate the performance of participants on different types of data and with different difficulty levels, and were able to manually set what a "correct" guess was. This allowed us to measure the performance of users on ground truth data. However, in our production system, the correctness of a guess would be determined by a machine learning system,

Figure 5.12: F_0 signal of N01 call quantized to a finite alphabet

perhaps combined with previous responses of other participants.

We recruited participants through a series of different online methods. The first method was to recruit users through a mailing list for graduate students and teachers of the computer science department at the University of Victoria. We recruited 9 participants in this manner, and gave them a short introduction to the Orchive project and to the game interface. We then sat with them for three levels of the game, where a level corresponds to a single user classification event. We then asked them to play for as long as they wanted, and measured the length of time they played for.

From this table, we can see that there was considerable variation in the length of time that different users played the game, from a low of about 11 minutes, to a maximum of about 47 minutes, and with a large standard deviation of about 11 minutes. The amount of classifications that they made also varied considerably, and was somewhat, but not heavily correlated with the amount of time they played the game. This was because some of the users listened to the sounds multiple times before making a selection, and others quickly went through the levels.

Participant ID	# classifications	% correct	length of time (h:m:s)
1	28	60.71	0:24:11
2	33	81.82	0:15:53
3	45	75.56	0:11:24
4	88	68.18	0:29:35
5	40	80.00	0:12:19
6	40	82.50	0:28:42
7	71	78.87	0:47:18
8	50	78.00	0:28:47
9	43	86.05	0:16:31
Mean	48.67	76.85	0:23:51
Median	43.00	78.87	0:24:11
Std. Dev.	19.09	7.86	0:11:23

Table 5.39:

The percent of correct classifications also varied considerably between users, with an average of about 77%, with a low of 60.71% and a high of 82.5%. Some levels were designed to considerably more difficult than others, and in a number of cases respondents were asked to differentiate between the same call type but vocalized by different matrilines, a task that even expert listeners can find challenging. From this initial pilot study, several minor interface flaws were identified and were fixed, including reducing the number of instruction screens describing how to use the game from four to one.

After this initial pilot study was completed, a series of announcements about the game were made on different online forums. The first was the same computer science mailing list for graduate students and teachers, but instead of asking for live volunteers, asking for people to follow a link and try the game. Similar announcements were posted to the authors Facebook feed, Google+ timeline and Twitter feed. In addition, invitations were sent out to a number of expert orca researchers that had participated in making annotations on the Orchive website. Another group of participants were recruited from the Orca-Live website (http://orca-live.net), a website from Orcalab that has a live audio stream of the audio from OrcaLab, and visited frequently by people with extensive experience of listening to orca vocalizations. The final group of participants were recruited using Google Adwords, in which a small ad budget of $40 was spent over the course of 5 days.

The emails to the experts was made on May 6th, and the posts to the social networks were done twice, on May 9th and on May 12th. From these efforts we

attracted a total of 633 unique visitors over the course of one month. The number of participants peaked strongly on the days the posts were made, which can be seen in Figure 5.13. Most people made only one visit to the site, but 177 people made two or more visits to the site, a histogram of the frequency of user visits can be seen in Figure 5.14 and. The average time that was spent on the site was 2 minutes 55 seconds, but some users spent considerably more time on the site, with a second peak in the engagement histogram at 3-10 minutes with 72 users in this histogram bin, shown in Figure 5.15. Of the 633 people who visited the front page of the site, 340 played at least one level of the game, which is a fairly high engagement ratio for an online game.

These results show that even this very simple game was quite engaging for certain participants with few actual game elements in the game. It would be of interest to look at these results when more game-like elements were added to the game, especially for users who were not previously interested in orca vocalizations.

There were a total of 122 levels created for the game using a custom Javascript interface, shown in Figure 5.16, which had functionality to allow the game designer to search for clips by call type, call matriline and recording. This interface was designed to allow researchers to create their own game levels for the purpose of training participants. The earlier static game interface that was originally built required a programmer to create levels using JSON, and from interactions with the researchers at OrcaLab, it was clear that an important feature would be to allow the researchers to create their own levels. This interface can also be used with clips automatically generated by a machine learning classifier to allow machine generated clips to be annotated by human experts.

The 122 levels in the game were designed to have a wide range of difficulty, some of the levels asked participants to distinguish between orcas and other marine mammals such as California Sea Lions *Zalophus californianus*, Pacific Whitesided Dolphins *Lagenorhynchus obliquidens* and Humpback Whales *Megaptera novaeangliae*. These types of levels were classified with very high accuracy, with the vast majority being classified at 100% accuracy.

Others asked the users to distinguish between different call types, for example level 1 which asked the user to distinguish between N01 and N03, N02 and N11, which gave a total classification accuracy across all users of 88%, but which the expert users classified with 100% accuracy.

The most difficult levels asked participants to classify the same call vocalized by

different matrilines, an example of this was a level that asked the participants to distinguish the N04 call of the A11 matriline from the N04 call of the A12 and A34 matrilines, a task on which the experts correctly identified 100% of the time but which all users classified correctly 72% of the time.

Surprisingly, one of the most difficult levels asked participants to classify orcas based on their pod, a task that experts typically found easy, but that untrained listeners found very difficult. One of these such levels asked participants to classify calls from the Southern Resident J pod against calls from the G pod and H pod, a task on which experts got 100% of the responses correct while untrained listeners got only 29% of the responses correct.

The percent of correct responses broken down by data source is shown in Table 5.40. From this, we can see that predictably, expert users were able to classify orca calls with the highest accuracy, with a combined average of 86.4% classified correctly. Participants recruited from Facebook had the second highest accuracy with 77.3%. Surprisingly, participants recruited from the Orca Live website had a lower average accuracy than those recruited from Facebook, with an average classification accuracy of 76.3%. These participants typically have great experience listening to orcas, and it would be expected that this familiarity with orca vocalizations would have given them a considerable advantage in classifying orca calls. One possible factor could be that participants coming from Facebook were also familiar with orca vocalizations, and that the process of recruitment on Facebook selected these people with higher probability than people not familiar with orca vocalizations.

The next highest average came from students recruited from the Computer Science, with an average accuracy of 60.8%. Users from Google+ had only a 25.8% accuracy, and those recruited from Google Ads and Twitter only had a 5% and 3% accuracy. These results are surprising, and could be due to issues with the instructions, as it was noted that several of the Google Ads users came from countries where English was not the native language, including China, Pakistan and Egypt. Clearly work on making the application internationalized is important when targeting these groups of participants.

Of the total of 3868 classifications of orca vocalizations, different amounts were classified by different groups of users. These results are shown in Table 5.41 and graphically in Figure 5.18. The largest number of classifications came from participants recruited from Facebook, with 855 classifications and the second most came from the Orca Live website, with 809 participants. These numbers were not surpris-

User group	Percent Correct
All	79.25
Expert	86.45
Students	60.75
Facebook	77.29
Twitter	3.07
Google+	25.84
OrcaLive	76.25
Google Adwords	5.45

Table 5.40: A table showing the average number of correct annotations assigned by different populations of users.

User Group	# Classifications
All	3868
Expert	678
Students	337
Facebook	855
Twitter	10
Google+	134
OrcaLive	809
Google Adwords	190
other	855

Table 5.41:

ing, as these two groups of participants contained people that were already interested in orca vocalizations. One surprising finding was that although only 9 experts were contacted, they contributed 678 classifications to the database. There were 337 classifications from students and professors from the computer science mailing list, 190 from Google Ad words, 134 from Google+, and 10 from Twitter.

One of the most striking graphs is the number of classifications per user. This is shown in Figure 5.19, and shows a sorted histogram of the number of classifications on the vertical axis and the participant ID on the horizontal axis. This graph can be interpreted as that there are few users who are highly engaged with the website and provide most of the classifications, and that most users only provide a small number of classifications. This behaviour has been seen in other crowdsourcing efforts, and highlights the importance of building interfaces that are tailored to these types of participants.

5.5 Expert Interface

It is vitally important to evaluate the system that we have created with the actual researchers who will be using this system. One could imagine using a variety of methods to evaluate this system, including Laboratory Experiments, Sample Surveys and Field Study.

We have completed a very small Ethnographic based Field Study in which several cetacean biologists made annotations using the Orchive interface. In this study, we observed 5 different scientists of varying levels of expertise, two of these were researchers with over 40 years of experience with orca vocalizations, one was a post-

graduate researcher who had done her Ph.D. studying orca vocalizations, one was a M.Sc. student who had not studied orca vocalizations before, but had studied pilot whales, and one was an undergraduate student who had extensive experience with visually observing whales.

The results from this field study were very interesting and encouraging. One of the respondents said:

"I wish I had this when I was doing my Ph.D. It would have made my research much easier"

another said:

"This makes it possible to finally access all this data easily."

When shown the predictions from machine learning, one of the researchers became quite excited and said:

"Wow! It can tell the difference between the different calls!"

One respondent had concerns that she would not know if the people who were making the annotations would really know what they were doing. This same sentiment was shared by most of the other respondents. Another discussion ensued about inter-observer reliability, the quality of annotations, and to know who had made which annotation, this highlighted the importance for reputation management systems to be implemented in this application.

Another topic that the subjects talked about was the importance for being able to temporarily hide data that would be used in an upcoming paper, so that other researchers would not be able to scoop them in the race to publication in scholarly journals. They said that:

"I want to share data with people in my lab, but not with people in other labs, at least until I publish"

Another comment that was made was:

"I want to be able to write my field notes and have them attached to a recording, but also be able to see all of them at once"

We must note that our Field Study borders on Critical Theory, because we are not just interested in studying these researchers, but in actively providing them tools to help them in their research. For this reason, it is difficult to remain completely objective about this research. In our opinion, it would be useful to have independent field studies carried out by dispassionate researchers.

However, the biggest lesson that was learned in this project was that in large part it was very difficult to engage expert biologists in using this interface. Upon further

reflection, this was clearly seen by Grudin [Gru88] where he asks researchers the question of who does the work and who gets the benefits. In this case, the person doing the work was the experts in Northern Resident Killer Whale vocalizations, and the people who got the benefit was the Machine Learning community, the Data Mining community, and potentially other future unknown researchers in orca communications.

Figure 5.13: A figure showing the number of visitors per day to the OrcaGame website. Four distinct peaks can be seen in this data, which correspond to the four different campaigns undertaken to recruit visitors. What is noticeable in this graph is that there was a long lasting tail of engagment of the game long after the initial recruitment pushes were done. This long tail shows that this game was very engaging for a small

Google Analytics

Go to this report

orchive.net - http://orchive.net
All Web Site Data [DEFAULT]

Frequency & Recency

Apr 27, 2013 - May 30, 2013

% of visits: 100.00%

Distribution

Count of Visits

Visits	Pageviews
800	**1,683**
% of Total: **100.00% (800)**	% of Total: **100.00% (1,683)**

Count of Visits	Visits	Pageviews
1	634	1,157
2	64	184
3	29	144
4	14	58
5	10	24
6	7	11
7	6	23
8	4	14
9-14	15	47
15-25	17	21

© 2013 Google

Figure 5.14: A histrogram showing the frequency of repeat visitors to the OrcaGame website. This histrogram shows that the majority of visitors only came to the site once, but that a substantial number of visitors (17) came back to the size between 15 and 25 times, which shows that for a certain subset of the population studied, this simple game was very engaging.

Google Analytics

Go to this report

orchive.net - http://orchive.net
All Web Site Data [DEFAULT]

Engagement

Apr 27, 2013 - May 30, 2013

% of visits: 100.00%

Distribution

Visit Duration

Visits	Pageviews
800	**1,683**
% of Total: **100.00% (800)**	% of Total: **100.00% (1,683)**

Visit Duration	Visits	Pageviews
0-10 seconds	**588**	**599**
11-30 seconds	**12**	**24**
31-60 seconds	**20**	**44**
61-180 seconds	**48**	**119**
181-600 seconds	**72**	**289**
601-1800 seconds	**38**	**278**
1801+ seconds	**22**	**330**

© 2013 Google

Figure 5.15: A histogram showing the length of time that visitors spent on the site. This shows that a large number of users (588) only briefly visited the site, but a substantial number (22) visited the site for longer than 30 minutes. This shows that certain participants found even this very simple version of game quite engaging.

Figure 5.16: A screenshot of the interface by which researchers can construct new custom levels of the orcagame using a web browser. All the annotations in the database are able to be added to the interface by the researcher, who can quickly build new levels to test the skill level of users.

Figure 5.17: A histogram showing the average percent correct for the different communities of users recruited to play the OrcaGame. In this histogram, the label "ex" refers to experts in orca vocalizations, "st" refers to students, "fb" to facebook, "tw" to twitter, "go" for Google+, "ol" to people recruited on the OrcaLive website and "ga" for users recruited from Google Ads.

Figure 5.18: A histogram showing the total number of annotations provided by different populations of users recruited to play the OrcaGame. In this histogram, the label "ex" refers to experts in orca vocalizations, "st" refers to students, "fb" to facebook, "tw" to twitter, "go" for Google+, "ol" to people recruited on the OrcaLive website and "ga" for users recruited from Google Ads.

Figure 5.19: A graph showing the total number of annotations per user, sorted by the amount of annotations per user. What one can see is a typical long-tail type graph, where a few users contribute many annotations, and most users provide only a few annotations.

Figure 5.20: A screenshot of the instructions from the OrcaGame. The original form of instructions had three separate screens, and this new screen was created from suggestions from members of the pilot study.

Figure 5.21: A screenshot of the OrcaGame interface, showing the query call at the top of the screen and the the reference calls below. the screen th

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The goal of this thesis was to produce a system that could allow researchers to annoate and classify all of the audio in a large bioacoustic collection.

In terms of making a system that allows researchers to listen to, view and annotate a large collection of audio, two systems were built in this thesis. The first was Orchive Version 1.0, and using this system, a group of 10 experts in orca vocalizations were able to create over 17,000 high quality annotations using this system, which is a proof of the ability of this system to enable scientists to use the system for annotations. This website has been accessed by over 3,900 people from 89 countries from around the world and in the last month saw 43 unique visitors. It is under active use by a number of orca researchers and in the last year has seen a dramatic increase in the number of people annotating recordings. A version of this system was used by a member of the research staff of the VENUS program and allowed him to find the first examples of marine mammal vocalizations in the audio data from the VENUS project. These results provide a demonstration of the success of this part of the system.

When building this system, several issues in terms of architecture, platform and user interface were identified and were addressed in the creation of Orchive Version 2.0, a completely rewritten framework that contains several thousand lines of code that has been made available as open source code on the Github website (http://github.com/openmir). This code consists of a central Python core that builds webpages with embedded Javascript apps and plugins which can be used separately from the main Orchive

website. The new web interface allows for the easy creation of annotations, the building of training and testing datasets, viewing of audio features, training of machine learning classifiers and viewing the results of these classification results. These new features also allow for the easy deployment of the Orchive website on new datasets, and indeed the new Orchive interface includes data from bird song recordings and partially annotated chant traditions. This new system was used to take the untrimmed annotations from experts in orca vocalizations and to create a set of trimmed annotations that were used in this thesis as the ORCAOBV dataset. The ability of the system to support activites such as this trimming, to see the results of classification experiments and to view audio features is a demonstration of the success of this new system.

In this thesis, a set of experiments using different sets of audio features and machine learning systems were conducted on a large set of training data, containing over 6000 seconds of audio in 11,041 clips containing the three labels orca/background/voice. The purpose of these experiments was to see if high classification accuracy could be obtained and if a trained classifier could be used to find all the orca vocalizations in the Orchive. The highest classification performance acheived was 92%, which is quite high, especially with the low Signal to Noise Ratio (SNR) in many of these clips. This high classification accuracy shows the success of this set of experiments.

Using a smaller subset of the data, a hybrid audio feature extraction and machine learning system that can run on a large cluster was developed. This system was used to segment all the audio in the Orchive, and took a total of approximately 2 days on 10 computers. This shows the success of the system in the task of classifying all the audio in the Orchive, which was a primary goal of this thesis.

In addition, experiments were undertaken to classify a set of 2985 orca vocalizations annotated into 12 call classes by a set of experts in orca vocalizations. A variety combinations of audio features and machine learning algorithms were used to classify this audio and a maximum classification accuracy of 74% was obtained. This is considerably higher than the 8% classification accuracy that would be attained if performance was random. The high classification accuracy in this task shows the success of this part of the thesis, and could be used in a system classify the entire set of clips from the segmented audio obtained by the hybrid bextract/sfplugin system.

Work done using a system similar to Symbolic Aggregate Approximation (SAX) to transform pitch contours into a string representation, which would allow the use of algorithms from bioinformatics to efficiently search the entire Orchive for repeated

patterns. The highest classfication accuracy for calls obtained with this system was 81%, which shows the potential utility of this algorithm for finding calls and other repeated patterns in the audio from the Orchive. This high classification accuracy shows the success of this part of the system described in this thesis.

With a call classification accuracy of around 80%, about 20% of the calls classified by the systems described above would be incorrectly labelled. To help correctly classify these calls, a system was developed to allow citizen scientists to help label the calls in the Orchive. This system has been integrated tightly into the Orchive Version 2.0 code, and allows researchers to have these citizen scientists help to correctly annotate these calls. The system that was developed took the form of a simple casual game using the common matching metaphor. An experiment was carried out to test this system, and over the course of 5 months, over 750 members of the public used this system and contributed 4899 annotations. 4203 of these annotations were done in a one month study period. This can be compared to the 5 years it took 10 experts to create 17,000 annotations, and shows the potential of large numbers of members of the public to quickly annotate large numbers of clips. Different populations of recruits had different classification accuracies, and for a set of student volunteers with no training in orca call recognition, a classification accuracy of 60.8%, people recruited from Facebook had an average classification accuracy of 77.3% and experts in orca calls had a classification of 86.4%. Given the difficult nature of the classification tasks in this hand constructed dataset, where participants were sometimes asked to classify calls based on matrilines, these results show the ability of the human perceptual system and of this citizen science interface to generate high quality call annotations. This result shows the success of this part of the system.

These results show that the objectives that were set out for this thesis have been achieved.

6.2 Future work

There remains considerable work to be done in regards to the Orchive as a whole. The first is to find more examples of clear, nearby, isolated orca vocalizations and background noise. There are 17,000 annotations, about half of which are orca vocalizations. A rough estimate would be that about 1 in 20 of these would be a loud, isolated vocalization, so one task we are working on is developing tools to quickly let us go through the existing annotations and to make training sets for Machine

Learning algorithms. In combination with this is the task to better trim the ends of orca calls to eliminate the silences that confound our Machine Learning algorithms. We are developing HTML5 based tools to help us to manually do this, and are also refining existing and developing new algorithms to automatically do this.

The most exciting directly applicable future work to the current work involves doing large scale clustering and vector quantization (VQ) on the loud, isolated, orca vocalizations found in this study. The VQ algorithm will give us as output an alphabet with an arbitrary sized dictionary. With this one dimensional quantized and clustered representation of the audio, we plan to use tools and techniques from Bioinformatics [SR02] and Symbolic Aggregate Approximation [LWea07] to extract information from this archive at a large scale. Although this project concentrated on classification, it is the clustering capabilities of Mahout that would be useful for this task, and would be even more appropriate for doing large scale clustering than for doing classification.

Chapter 7

Web Links

The following are web links to the various web-based tools developed in the course of this research:

Orchive Version 1.0: `http://orchive.cs.uvic.ca`

Orchive Version 2.0: `http://orchive.net`

Chapter 8

Publications from this Research

Presented here is a list of all the publications that come from the research presented in this thesis:

The Orchive : Data mining a massive bioacoustic archive, Steven Ness, Helena Symonds, Paul Spong, George Tzanetakis. ICML 2013 Workshop on Machine Learning for Bioacoustics

Sattar, F., Driessen, P. F., Tzanetakis, G., Ness, S. R., and Page, W. H. (2011, August). Automatic event detection for long-term monitoring of hydrophone data. In Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on (pp. 668-674). IEEE.

van Kranenburg, P., Biro, D. P., Ness, S. R., and Tzanetakis, G. (2011). A Computational Investigation of Melodic Contour Stability in Jewish Torah Trope Performance Traditions. In ISMIR (pp. 163-168).

Ness, Steven R., Alex Lerch, and George Tzanetakis. Strategies for orca call retrieval to support collaborative annotation of a large archive. Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on. IEEE, 2011.

Steven R. Ness, Daniel Peter Biro and George Tzanetakis Computer-assisted cantillation and chant research using content-aware web visualization tools, Multimedia Tools and Applications, Accepted for publication

Steven R. Ness, Daniel Peter Biro, George Tzanetakis : Content-aware web browsing and visualization tools for cantillation and chant research, 7th International Workshop on Content-Based Multimedia Indexing

Steven R. Ness, Matthew Wright, Luis Gustavo Martins, George Tzanetakis: Chants and Orcas: semi-automatic tools for audio annotation and analysis in niche domains. ACM Multimedia 2008: 9-16

Daniel Peter Biro, Steven Ness, Matthew Wright, W. Andrew Schloss and George Tzanetakis Decoding the Song: Histogram-Based Paradigmatic and Syntagmatic Analysis of Melodic Formulae in Hungarian Laments, Jewish Torah Trope, Tenth Century Plainchant and Koran Recitation EMUS Expressivity in MUsic and Speech : IRCAM - Institut de Recherche et de Coordination Acoustique/Musique - Paris, France

Chapter 9

Glossary

AJAX - Asynchronous Javascript and XML - A technique and set of technology used in building Javascript based web interfaces in which XML (or other data) is transferred from a server to a client. It allows the client to only update small sections of a web page and not have to refresh the entire page as is common in traditional web interfaces.

API - Application Programming Interface. A set of interfaces which programs use to communciate with each other.

Backbone - A Javascript framework for the development of Model/View/Controller apps on the client. It enables the development of large and feature rich client side applications, and was used as the primary Javascript framework in this thesis for developing the Orchive Version 2.0 interface.

Bootstrap - A HTML5/CSS framework developed by Twitter that allows for easy development of attractive web interfaces using a grid layout metaphor. Used as the primary layout tool in the Orchive Version 2.0 interface for layout.

Celery - Distributed task system used to run jobs in an asynchronous manner using worker nodes. Used in this thesis to allow audio feature extraction and machine learning jobs to run in asynchronously on a single machine or on a cluster of machines.

Cross-validation (Computer Science) - A method used to test the accuracy of machine learning classifiers by separating a dataset into examples used to train the

classifier and examples used to test the classifier.

DSP - Digital Signal Processing. Digital techniques which can be used to process audio.

Flux (MIR) - In this work, is the norm of the difference vector between two successive magnitude/power spectra.

iOS - A mobile operating system developed by Apple to run on the iPod, iPhone and iPad families. iOS is one of the possible targets of the PhoneGap platform. In this thesis, the primary mobile platform used by the citizen science games was iOS.

JSON - Javascript Object Notation. A modern data-exchange format, used extensively as a way for servers to send data to web-based clients.

MVC - Model/View/Controller. A design pattern for building systems that interact with users and store data in a database.

Memcached - A memory based key-value store. Memcached is used in the system described by this thesis to cache images of spectrograms in order to enhance the speed of the web interface.

Mysticeti - A suborder of cetaceans of which one of the distinguishing features is they use baleen to filter prey from the water, in contrast to the Odontocetes, which have teeth.

MFCC coefficients (MIR) - A way to transform a standard spectrum into one that more closely approximates how the human ear perceives sound.

Music Information Retrieval (Computer Science) - also known by the acronym MIR, a new field of study where one applies tools from areas such as Digital Signal Processing, Audio Feature Extraction and Machine Learning to help people understand and retrieve information from music or audio.

Odontocete - A suborder of cetaceans of which one of the distinguishing features is they have teeth, in contrast to the Mysticeti, which have baleen.

OrcaLab - A land based research station founded by Dr. Paul Spong in 1970 to study orca vocalizations and behaviour. It is run by Dr. Paul Spong and Helena Symonds with assistance from volunteers. All the data in the Orchive was recorded by the OrcaLab team.

Phonic Lips - Also known as “museau de singe” or “monkey lips” - The mechanism by which odontocetes produce sound.

PhoneGap - A software package that allows the deployment of HTML5/Javascript based interfaces on tablets and smartphones such as the iPad and iPhone. PhoneGap was used in this thesis to deploy the citizen science based casual game to mobile devices.

Redis - A fast, memory based key-value store often that is often used as a cache to store previously calculated results. It is also used to as a central repository to store the results of calculations performed by the celery distributed task system.

REST - Representational State Transfer. A method in which a server and a client can communicate over HTTP to transmit state between the two of them.

Rolloff (MIR) - A measure of the steepness of falloff in an audio spectrum

Self-Organizing Map (Computer Science) - A technique that maps a high dimensional feature space to a lower space. It is a similar technique to artificial neural networks.

Spectral Centroid (MIR) - A measure of the “center of mass” of a spectrum.

XML (Computer Science) - eXtensible Markup Language - A popular tree-tree based format for encoding data

Bibliography

[ABD+97] K.L. Atkins, D.E. Brownlee, T. Duxbury, C.-W. Yen, P. Tsou, and J.M. Vellinga. Stardust: Discovery's interstellar dust and cometary sample return mission. In *STARDUST: Discovery's InterStellar dust and cometary sample return mission*, volume 4, pages 229–245, 1997.

[Ada06] O. Adam. Advantages of the Hilbert Huang transform for marine mammals signals analysis. *J. Acoust. Soc. Am.*, 120:2965–2973, Nov 2006.

[AFHA04] W. W. Au, J. K. Ford, J. K. Horne, and K. A. Allman. Echolocation signals of free-ranging killer whales (Orcinus orca) and modeling of foraging for chinook salmon (Oncorhynchus tshawytscha). *J. Acoust. Soc. Am.*, 115:901–909, Feb 2004.

[Aria] Aristotle. The history of animals, book 4, part 9.

[Arib] Aristotle. On the soul, book 2, chapter 8.

[BBB+10] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, volume 4, 2010.

[BD82] Ford J. K. B. and Fisher H. D. Killer whale (orcinus orca) dialects as an indicator of stocks in british columbia. In *Int. Whaling Comm. Rep. Comm*, volume 32, pages 671 –679, 1982.

[BHS+09] Stan Boutin, Diane L Haughland, Jim Schieck, Jim Herbers, and Erin Bayne. A new approach to forest biodiversity monitoring in canada. *Forest Ecology and Management*, 258:S168–S175, 2009.

[BLFH96] LANCE G. BARRETT-LENNARD, JOHN K.B. FORD, and KATHY A. HEISE. The mixed blessing of echolocation: differences in sonar use by fish-eating and mammal-eating killer whales. *Animal Behaviour*, 51(3):553 – 565, 1996.

[BM02] Erin Bradner and Gloria Mark. Why distance matters: effects on cooperation, persuasion and deception. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 226–235, 2002.

[BM06] A.J. Westphal B.M. Mendez, N. Craig. Stardust@home: Enlisting students and the public in the search for interstellar dust. In *Stardust@home: Enlisting Students and the Public in the Search for Interstellar Dust*, 2006.

[BM07] J. C. Brown and P. J. Miller. Automatic classification of killer whale vocalizations using dynamic time warping. *J. Acoust. Soc. Am.*, 122:1201–1207, Aug 2007.

[BMEWL11] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, pages 591–596, 2011.

[BOE+90] M.A. Bigg, P.F. Olesiuk, G.M. Ellis, J.K.B. Ford, and K.C. Balcomb III. Social organization and genealogy of resident killer whales (orcinus orca) in the coastal waters of british columbia and washington state. Technical report, UBC, 1990.

[Bra08] Daren C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–90, 2008.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BS91] Liam J. Bannon and Kjeld Schmidt. Cscw: Four characters in search of a context. In *Studies in Computer Supported Cooperative Work*, volume 8 of *Human Factors in Information Technology*, pages 3–17. North-Holland, 1991.

[BT05] Stuart Bray and George Tzanetakis. Distributed audio feature extraction for music. In *ISMIR*, pages 434–437, 2005.

[BTL12] Luke Barrington, Douglas Turnbull, and Gert Lanckriet. Game-powered machine learning. *Proceedings of the National Academy of Sciences*, 109(17):6411–6416, 2012.

[BWC08] R. Bardeli, D. Wolff, and M. Clausen. Bird song recognition in complex audio scenes. In *Proceedings of the International Expert meeting on IT-based detection of bioacoustical patterns*, BfN-Skripten 234, pages 93–101. Bundesamt für Naturschutz, 2008.

[BWK+10] R. Bardeli, D. Wolff, F. Kurth, M. Koch, K.H. Tauchert, and K.H. Frommolt. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recognition Letters*, 31(12):1524–1534, 2010.

[CA06] C. W. Clark and N. S. Altman. Acoustic detections of blue whale (balaenoptera musculus) and fin whale (b. physalus) sounds during a surtass lfa exercise. *Journal of Oceanic Engineering*, 31:120–128, 2006.

[CAN96] T W Cranford, M Amundin, and K S Norris. Functional morphology and homology in the odontocete nasal complex: implications for sound generation. *J Morphol*, 228(3):223–85, 1996.

[CH85] Cummings W. C. and D. V. Holliday. Passive acoustic location of bowhead whales in a population census off point barrow, alaska. *The Journal of the Acoustical Society of America*, 78:1163–1169, 1985.

[CH92] S. Le Cessie and J. C. Van Houwelingen. Ridge Estimators in Logistic Regression. *Applied Statistics*, 41(1):191–201, 1992.

[CL01] C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[CLS10] Chris Cannam, Christian Landone, and Mark B. Sandler. Sonic visualiser: an open source application for viewing, analysing, and annotating music audio files. In *ACM Multimedia*, pages 1467–1468, 2010.

[Cra00] T. Cranford, T. W. Cranford. *In search of impulse sound sources in Odontocetes*, pages 109–155. Springer, New York, 2000.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[DA82] Marilyn E. Dahlheim and Frank Awbrey. A classification and comparison of vocalizations of captive killer whales (orcinus orca). *J. Acoust. Soc. Am*, 72:661–670, 1982.

[dCK02] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[Dee] Volker B. Deecke. Studying marine mammal cognition in the wild: A review of four decades of playback experiments. *Aquatic mammals*, 32:461–482.

[DFS00] V.B. Deecke, J.K.B. Ford, and P. Spong. Dialect change in resident killer whales (orcinus orca): implications for vocal learning and cultural transmission. *Animal Behaviour*, 60(5):619–638, 2000.

[DJ06] V. B. Deecke and V. M. Janik. Automated categorization of bioacoustic signals: avoiding perceptual pitfalls. *J. Acoust. Soc. Am.*, 119:645–653, 2006.

[DRGOH+09] L. Doksaeter, O. Rune Godo, N. Olav Handegard, P. H. Kvadsheim, F. P. Lam, C. Donovan, and P. J. Miller. Behavioral responses of herring (Clupea harengus) to 1-2 and 6-7 kHz sonar signals and killer whale feeding sounds. *J. Acoust. Soc. Am.*, 125:554–564, Jan 2009.

[DT96] Shlomo Dubnov and Naftali Tishby. Testing for gaussianity and non linearity in the sustained portion of musical sounds., 1996.

[EDSW09] J. A. Estes, D. F. Doak, A. M. Springer, and T. M. Williams. Causes and consequences of marine mammal population declines in southwest Alaska: a food-web perspective. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 364:1647–1658, Jun 2009.

[FBC08] K.H. Frommolt, R. Bardeli, and M. Clausen. Computational bioacoustics for assessing biodiversity. In *Proceedings of the International Expert meeting on IT-based detection of bioacoustical patterns, BfN-Skripten*, number 234. Citeseer, 2008.

[FCH+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[FD02] J. Futrelle and S. Downie. Interdisciplinary communities and research issues in music information retrieval. In *Proc. Int. Conf. on Music Information Retrieval(ISMIR)*, 2002.

[Fel94] Steven Feld. From ethnomusicology to echo-muse-ecology: Reading r. murray schafer in the papua new guinea rainforest. *The Soundscape Newsletter*, 8:4–6, 1994.

[FF83] J.K.B. Ford and H.D. Fisher. *Group-specific dialects of killer whales (Orcinus orca) in British Columbia*, pages 129–161. Westview Press, Boulder, Colorado, 1983.

[Fie09] Christopher A Fiebrich. History of surface weather observations in the united states. *Earth-Science Reviews*, 93(3):77–84, 2009.

[Fis06] Gerhard Fischer. Distributed intelligence: extending the power of the unaided, individual human mind. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 7–14, New York, NY, USA, 2006. ACM.

[FMB99] Norris T. F., M. McDonald, and J. Barlow. Acoustic detections of singing humpback whales (megaptera novaeangliae) in the eastern north pacific during their northbound migration. *The Journal of the Acoustical Society of America*, 106(1):506–514, 1999.

[FN92] G. Fischer and K. Nakakoji. "beyond the macho approach of artificial intelligence: Empower human designers - do not replace them,". *Knowledge-Based Systems Journal, Special Issue on AI in Design*, 5(1):15–30, 1992.

[FNO+98] Gerhard Fischer, Kumiyo Nakakoji, Jonathan Ostwald, Gerry Stahl, and Tamara Sumner. Embedding critics in design environments. *Readings in intelligent user interfaces*, pages 537–561, 1998.

[FOH04] A. D. Foote, R. W. Osborne, and A. R. Hoelzel. Environment: whale-call response to masking boat noise. *Nature*, 428:910, 2004.

[For87] J.K.B. Ford. A catalogue of underwater calls produced by killer whales (orcinus orca) in british columbia. Technical Report 633, Canadian Data Report of Fisheries and Aquatic Science, 1987.

[For89] J.K.B. Ford. Acoustic behaviour of resident killer whales (orcinus orca) off vancouver island, british columbia. *Canadian Journal of Zoology*, 64:727–745, 1989.

[For91] John KB Ford. Vocal traditions among resident killer whales (orcinus orca) in coastal waters of british columbia. *Canadian Journal of Zoology*, 69(6):1454–1483, 1991.

[Gru88] Jonathan Grudin. Why cscw applications fail: problems in the design and evaluationof organizational interfaces. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 85–93, 1988.

[Hal13] Xanadu Halkias, editor. *ICML 2013 Workshop on Machine Learning for Bioacoustics*, 2013.

[HHK00] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7(2):174–196, 2000.

[HK07] Danoush Hosseinzadeh and Sridhar Krishnan. Combining vocal source and mfcc features for enhanced speaker recognition performance using gmms. In *MMSP*, pages 365–368, 2007.

[HNV+09] M. M. Holt, D. P. Noren, V. Veirs, C. K. Emmons, and S. Veirs. Speaking up: Killer whales (Orcinus orca) increase their call amplitude in response to vessel noise. *J. Acoust. Soc. Am.*, 125:27–32, Jan 2009.

[HO86] A.R. Hoelzel and R.W. Osborne. *Killer whale call characteristics: implications for cooperative foraging strategies*, pages 373–403. A.R. Liss, New York, 1986.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[How08] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Business, 2008.

[HP04] Lu Hong and Scott E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences*, 101(46):16385–16389, 2004.

[Hut80] Edwin. Hutchins. *Culture and inference : a Trobriand case study*. Harvard University Press, Cambridge, Mass., 1980.

[Hut96] Edwin Hutchins. *Cognition in the wild*. MIT Press, 1996.

[IEE00] IEEE. *Robust pitch tracking for prosodic modeling in telephone speech*, volume 3, 2000.

[Jan99] Vincent M. Janik. Pitfalls in the categorization of behaviour: a comparison of dolphin whistle classification methods. *Animal Behaviour*, 57(1):133–143, January 1999.

[Jan13] Vincent M Janik. Cognitive skills in bottlenose dolphin communication. *Trends in cognitive sciences*, 2013.

[JC.08] Brown JC. Mathematics of pulsed vocalizations with application to killer whale biphonation. *J Acoust Soc Am.*, 123(5):2875–83, 2008.

[JD94] G. Janik, V. M. Dehnhardt and Todt D. Signature whistle variations in a bottlenosed dolphin, tursiops truncatus. *Behav. Ecol. Sociobiol*, 35:243–248, 1994.

[JFB00] G.E. Ellis J.K.B. Ford and K.C. Balcomb. *Killer Whales 2nd ed*. University of British Columbia Press, Vancouver, 2000.

[JHSS93] Gaetz W. Jantzen, Weinberg H., P. Spong, and H. Symonds. A neural network method for recognition of individual orcinus orca based on their acoustic behaviour: phase 1. In *OCEANS '93. Engineering in Harmony with Ocean. Proceedings*, 1993.

[JKSW13] Vincent M. Janik, Stephanie L. King, Laela S. Sayigh, and Randall S. Wells. Identifying signature whistles from recordings of groups of unrestrained bottlenose dolphins (tursiops truncatus). *Marine Mammal Science*, 29(1):109–122, 2013.

[JL95] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[JM06] J.C.Brown and P.J.O. Miller. Classifying killer whale vocalization using time warping. *Acoustics Today*, July 2006.

[JS98] Vincent M Janik and Peter J.B Slater. Context-specific use suggests that bottlenose dolphin signature whistles are cohesion calls. *Animal Behaviour*, 56(4):829 – 838, 1998.

[JS00] V.M. Janik and P.J.B. Slater. The different roles of social learning in vocal communication. *Animal Behaviour*, 60:1–11, 2000.

[KCS08] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456, New York, NY, USA, 2008. ACM.

[KD06] Anssi Klapuri and Manuel Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006.

[KGM97] H. Khanna, S.L.L. Gaunt, and D.A. McCallum. Digital spectrographic cross-correlation: tests of sensitivity. *Bioacoustics*, 7:209–234, 1997.

[Kin12] Stephanie L King. *The vocal imitation of bottlenose dolphin (Tursiops truncatus) signature whistles: their use in vocal matching interactions and their role as vocal labels*. PhD thesis, University of St Andrews, 2012.

[KL03] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[KM96] Donald E Kroodsma and Edward H Miller. *Ecology and evolution of acoustic communication in birds*. Comstock Pub., 1996.

[KSRR12] Vinh Ngoc Khuc, Chaitanya Shivade, Rajiv Ramnath, and Jay Ramanathan. Towards building large-scale distributed systems for twitter

sentiment analysis. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 459–464. ACM, 2012.

[KSW+13] Stephanie L. King, Laela S. Sayigh, Randall S. Wells, Wendi Fellner, and Vincent M. Janik. Vocal copying of individually distinctive signature whistles in bottlenose dolphins. *Proceedings of the Royal Society B: Biological Sciences*, 280(1757), 2013.

[LBF06] Beinan Li, John Ashley Burgoyne, and Ichiro Fujinaga. Extending audacity for audio annotation, 2006.

[LCN+09] Geoffrey S LeBaron, Richard J Cannings, Daniel K Niven, Gregory S Butcher, G Thomas Bancroft, Paul W Sykes Jr, Sarah McCarn Elliott, Noah Strycker, and Peter Read. The 109th christmas bird count. *American Birds*, 63:2–7, 2009.

[Lin58] C. Linnaeus. *Tomus I. Systema naturae per regna tria naturae, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis..[1-4], 1-824*. Editio decima, reformata. Holmiae. (Laurentii Salvii), 1758.

[Liu07] Y. Q. Liu. Macaulay library: A case study of the library of natural sound. *Digital Libraries Forum 12*, 2007.

[LKLC03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM.

[LLS04] Watwood S. L., Tyack P. L., and Wells R. S. Whistle sharing in paired male bottlenose dolphins, tursiops truncatus. *Behav. Ecol. Sociobiol*, 55:531–543, 2004.

[LMHFJ10] J.P. Luke, J.G. Marxichal-Hernandez, F.Rosa, and J.Almunia. Real time automatic detection of orcinus orca vocalizations in a controlled environment. *Applied Acoustics*, 71(8):771 – 776, 2010.

[LNK87] M. Lahat, R. Niederjohn, and D. Krubsack. A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(6):741–750, 1987.

[LO06] Tao Li and M. Ogihara. Toward intelligent music information retrieval. *Multimedia, IEEE Transactions on*, 8(3):564–574, 2006.

[Log00] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*, 2000.

[LQ11] Zhu Le-Qing. Insect sound recognition based on mfcc and pnn. In *Proc. of the 2011 Int. Conf. on Multimedia and Signal Processing*, CMSP '11, pages 42–46, 2011.

[LSYS07] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Jung-Mau Su. Automatic music genre classification using modulation spectral contrast feature. In *ICME*, pages 204–207, 2007.

[LvA09] Edith Law and Luis von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proc. CHI 2009*, pages 1197–1206, 2009.

[LWea07] Jessica Lin, Li Wei, and et al. Experiencing sax: a novel symbolic representation of time series, 2007.

[Lyo82] R. Lyon. A computational model of filtering, detection, and compression in the cochlea. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82.*, volume 7, pages 1282 – 1285, may 1982.

[Lyo90] R. F. Lyon. Automatic gain control in cochlear mechanics. In P Dallos et al., editor, *The Mechanics and Biophysics of Hearing*, pages 395–420. Springer-Verlag, 1990.

[MB00a] P. J. Miller and D. E. Bain. Within-pod variation in the sound production of a pod of killer whales, Orcinus orca. *Anim Behav*, 60:617–628, Nov 2000.

[MB00b] P.J.O. Miller and D.E. Bain. Within-pod variation in killer whale calls. *Animal Behaviour*, 60:617–628, 2000.

[McC95] Brenda McCowan. A new quantitative technique for categorizing whistles using simulated signals and whistles from captive bottlenose dolphins (delphinidae, tursiops truncatus). *Ethology*, 100(3):177–193, 1995.

[McG95] Joseph E. McGrath. Methodology matters: doing research in the behavioral and social sciences. *Human-computer interaction: toward the year 2000*, pages 152–169, 1995.

[MD02] Dominic Mazzoni and Roger B. Dannenberg. A fast data structure for disk-based audio editing. *Comput. Music J.*, 26(2):62–76, 2002.

[ME05] Michael Mandel and Daniel Ellis. Song-level features and support vector machines for music classification. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[Mil02] Patrick J. O. Miller. Mixed-directionality of killer whale stereotyped calls: A direction of movement cue? *Behavioral Ecology and Sociobiology*, 52(3):262–270, 2002.

[Mil06] P. J. Miller. Diversity in sound pressure levels and estimated active space of resident killer whale vocalizations. *J. Comp. Physiol. A Neuroethol. Sens. Neural. Behav. Physiol.*, 192:449–459, May 2006.

[MMF06] Daniel McEnnis, Cory McKay, and Ichiro Fujinaga. Overview of omen. In *Proceedings of the International Conference on Music Information Retrieval*, 2006.

[MNS+99] B.S. Mulder, B.R. Noon, T.A. Spies, M.G. Raphael, C.J. Palmer, A.R. Olsen, G.H. Reeves, and H.H. Welsh. The strategy and design of the effectiveness monitoring program for the northwest forest plan. Technical report, 1999.

[MS02] A. B. Morton and H. K. Symonds. Displacement of orcinus orca (l.) by high amplitude sound in british columbia. *Canada ICES Journal of Marine Science*, 59:71–80, 2002.

[MSP07] Patrick J O Miller, Filipa I P Samarra, and Aurlie D Perthuison. Caller sex and orientation influence spectral characteristics of two-voice stereotyped calls produced by free-ranging killer whales. *J Acoust Soc Am*, 121(6):3932–7, 2007.

[MZSM04] P N. Manley, W. J. Zielinski, M. D. Schlesinger, and S. R. Mori. Evaluation of a multiple-species approach to monitoring species at the ecoregional scale. *Ecological Applications*, 14:296–310, 2004.

[NMW98] Bonnie A. Nardi, James R. Miller, and David J. Wright. Collaborative, programmable intelligent agents. *Commun. ACM*, 41(3):96–104, 1998.

[Now05] Douglas P Nowacek. Acoustic ecology of foraging bottlenose dolphins (tursiops truncatus), habitat-specific use of three sound types. *Marine mammal science*, 21(4):587–602, 2005.

[NRH+99] S. Nummela, T. Reuter, S. Hemil?, P. Holmberg, and P. Paukku. The anatomy of the killer whale middle ear (Orcinus orca). *Hear. Res.*, 133:61–70, Jul 1999.

[NSST13] Steven Ness, Helena Symonds, Paul Spong, and George Tzanetakis. The orchive : Data mining a massive bioacoustic archive. In *ICML 2013 Workshop on Machine Learning for Bioacoustics*, 2013.

[NSWM06] A. E. Nousek, P. J. Slater, C. Wang, and P. J. Miller. The influence of social affiliation on individual vocal signatures of northern resident killer whales (Orcinus orca). *Biol. Lett.*, 2:481–484, 2006.

[NWMT08] Steven R. Ness, Matthew Wright, L. Gustavo Martins, and George Tzanetakis. Chants and orcas: semi-automatic tools for audio annotation and analysis in niche domains. In *Proceedings of the 2nd ACM workshop on Multimedia semantics*, MS '08, pages 9–16. ACM, 2008.

[OADF11] Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. *Mahout in action*. Manning, 2011.

[OBN03] Julie N. Oswald, Jay Barlow, and Thomas F. Norris. Acoustic identification of nine delphinid species in the eastern tropical pacific ocean. *Marine Mammal Science*, 19(1):20–037, 2003.

[OO00] G.M. Olson and J.S. Olson. Distance matters. *Human-Computer Interaction*, 15(2/3):139–178, 2000.

[oO11] The Cornell Lab of Ornithology. *Raven Pro: Interactive Sound Analysis Software (Version 1.4) [Computer software]*, 2011. Software available at http://www.birds.cornell.edu/raven.

[PLS04] Sofie M. Van Parijs, Teo Leyssen, and Tiu Simil. Sounds produced by norwegian killer whales, orcinus orca, during capture. *J. Acoust. Soc. Am.*, 116:557–560, 2004.

[PMS04] P.L. Tyack P.J.O. Miller, A.D. Shapiro and A.R. Solow. Call-type matching in vocal exchanges of free-ranging resident killer whales, orcinus orca. *Animal Behaviour*, 67:1099–1107, 2004.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Qui93] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Red98] Johan Redström. ?is acoustic ecology about ecology. In *Reflections on the international conference on acoustic ecology ?Stockholm, hey listen*, 1998.

[RHW02] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1:213, 2002.

[Rie93] K. Riede. Monitoring biodiversity: analysis of amazonian rainforest sounds. *Ambio*, pages 546–548, 1993.

[RL93] Buck J. R. and Tyack P. L. A quantitative measure of similarity for tursiops truncatus signature whistles. *J. Acoust. Soc. Am.*, 94:2497–2506, 1993.

[RRK+90] Dennis W Ruck, Steven K Rogers, Matthew Kabrisky, Mark E Oxley, and Bruce W Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *Neural Networks, IEEE Transactions on*, 1(4):296–298, 1990.

[Sal97] Gavriel Salomon. *Distributed cognitions: Psychological and educational considerations*. Cambridge University Press, 1997.

[SB08] T. Scott Brandes. Automated sound recording and analysis techniques for bird surveys and conservation. *Bird Conservation International*, 18(S1):S163–S173, 2008.

[SBHK97] Tamara Sumner, Nathalie Bonnardel, and Benedikte Harstad Kallak. The cognitive ergonomics of knowledge-based design support systems. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–90, New York, NY, USA, 1997. ACM.

[SFC98] K. M. Stafford, C. G. Fox, and D. S. Clark. Longrange acoustic detection and localization of blue whale calls in the northeast pacific ocean. *The Journal of the Acoustical Society of America*, 104:3616–3625, 1998.

[Sha06] A. D. Shapiro. Preliminary evidence for signature vocalizations among free-ranging narwhals monodon monoceros. *J. Acoust. Soc. Am.*, 120:1695–1705, 2006.

[SLB+08] Anze Slosar, Kate Land, Steven Bamford, Chris Lintott, Dan Andreescu, Phil Murray, Robert Nichol, Jordan M. Raddick, Kevin Schawinski, Alex Szalay, Daniel Thomas, and Jan Vandenberg. Galaxy zoo: Chiral correlation function of galaxy spins. *Mon. Not. R. Astron. Soc*, Sep 2008.

[SM97] Ben Schneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, 1997.

[Smi04] Bruce R Smith. Listening to the wild blue yonder: The challenges of acoustic ecology. *Hearing Cultures: Essays on Sound, Listening and Modernity*, pages 21–41, 2004.

[SP67] Richard C. Singleton and T.C. Poulter. Spectral analysis of the call of the male killer whale. *Audio and Electroacoustics, IEEE Transactions on*, 15(2):104–113, 1967.

[SPHD08] Jerome Sueur, Sandrine Pavoine, Olivier Hamerlynck, and Stephanie Duvail. Rapid acoustic survey for biodiversity appraisal. *PLoS ONE*, 3, 12 2008.

[SR02] I.N. Sarkar and T.C. Rindflesch. Discovering protein similarity using natural language processing. *Proc AMIA Symp*, pages 677–81, 2002.

[SSB+97] Bernhard Scholkopf, Kah Kay Sung, Christopher J. C. Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765, 1997.

[Str95] H. Strager. Pod-specific call repertoires and compound calls of killer whales, orcinus orca linnaeus, 1758, in the waters of northern norway. *Canadian Journal of Zoology*, 73:1037–1047, 1995.

[STWS90] Laela S Sayigh, Peter L Tyack, Randall S Wells, and Michael D Scott. Signature whistles of free-ranging bottlenose dolphins tursiops truncatus: stability and mother-offspring comparisons. *Behavioral Ecology and Sociobiology*, 26(4):247–260, 1990.

[Sur05] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

[SW09] Ari D. Shapiro and Chao Wang. A versatile pitch tracking algorithm: From human speech to killer whale vocalizations. *The Journal of the Acoustical Society of America*, 126(1):451–459, 2009.

[TLSS07] G. Tzanetakis, M. Lagrange, P. Spong, and H. Symonds. Orchive: Digitizing and analyzing orca vocalizations. In *Proc. of the RIAO, Large-Scale Semantic Access to Content Conference*. RIAO, 2007.

[Tra08] John Travis. Science and commerce: Science by the masses. *Science*, 319(5871):1750–1752, 2008.

[Tru01] Barry Truax. Handbook of acoustic ecology, (cd-rom version). *Computer Music Journal*, 25:93–94, 2001.

[TSH94] K. W. Thornton, G. E. Saul, and D. E. Hyatt. Environmental monitoring and assessment program assessment framework. Technical report, 1994.

[Tza08] G. Tzanetakis. *Marsyas-0.2: A case study in implementing music information retrieval systems*, chapter 2, pages 31–49. Intelligent Music

Information Systems: Tools and Methodologies. Information Science Reference, 2008. Shen, Shepherd, Cui, Liu (eds).

[UCKP09] I.R. Urazghildiiev, C.W. Clark, T.P. Krein, and S.E. Parks. Detection and recognition of north atlantic right whale contact calls in the presence of ambient noise. *Oceanic Engineering, IEEE Journal of*, 34(3):358–368, 2009.

[UK99] N.S. Urquhart and T.M. Kincaid. Designs for detecting trend from repeated surveys of ecological resources. *Journal of Agricultural, Biological, and Environmental Statistics*, pages 404–414, 1999.

[UPL98] N. S. Urquhart, S. G. Paulsen, and D. P. Larsen. Monitoring for policy-relevant regional trends over time. *Ecological Applications*, pages 246–257, 1998.

[vAD04] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM.

[vAD08] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[VAV11] Deecke VB, Foote AD, and Janik VM. Vocal behaviour and feeding ecology of killer whales orcinus orca around shetland, uk. *Aquatic Biology*, 13:79–88, 2011.

[VDS99] J.K.B. Ford V.B. Deecke and P. Spong. Quantifying complex patterns of bioacoustic variation: use of a neural network to compare killer whale (orcinus orca) dialects. *Journal of the Acoustical Society of America*, 105:2499–2507, 1999.

[W06] Cranford T W. Odontocete nasalizations: Morphology, physiology, and evolution of sonar signal generation in toothed whales,. *J. Acoust. Soc. Am.*, 120:3189–3190, 2006.

[Wat67] W. A Watkins. *The harmonic interval: fact or artifact in spectral analysis of pulse trains*, volume 2, pages 15–43. Pergamon New York, 1967.

[Wes02] Hildegard Westerkamp. Linking soundscape composition and acoustic ecology. *Organised Sound*, 7(1):51–56, 2002.

[WF05] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[WLSS06] Brigitte M. Weiss, Friedrich Ladich, Paul Spong, and Helena Symonds. Vocal behavior of resident killer whale matrilines with newborn calves: The role of family signatures. *The Journal of the Acoustical Society of America*, 119(1):627–635, 2006.

[WN04] I. Watson and P. Novelly. Making the biodiversity monitoring system sustainable: Design issues for large-scale monitoring systems. *Austral Ecology*, pages 16–30, 2004.

[WOTW05] Stephanie L. Watwood, Edward C.G. Owen, Peter L. Tyack, and Randall S. Wells. Signature whistle use by temporarily restrained and free-swimming bottlenose dolphins, tursiops truncatus. *Animal Behaviour*, 69(6):1373 – 1386, 2005.

[Wri00] Kendall Wrightson. An introduction to acoustic ecology. *Soundscape: The journal of acoustic ecology*, 1(1):10–13, 2000.

[WTP+10] J. Wimmer, M. Towsey, B. Planitz, P. Roe, and I. Williamson. Scaling acoustic data analysis through collaboration and automation. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 308 –315, dec. 2010.

[WWLA00] Richard R. Fay Whitlow W. L. Au, Arthur N. Popper. *Hearing by whales and dolphins*. Shar, 2000.

[YFM+10] Harald Yurk, Olga Filatova, Craig O. Matkin, Lance G. Barrett-Lennard, and Michael Brittain. Sequential habitat use by two resident killer whale (orcinus orca) clans in resurrection bay, alaska, as determined by remote acoustic monitoring. *Aquatic Mammals*, 36:67–78, 2010.

[Yur05] Harald Yurk. *The evolutionary history of resident killer whale clans in the northeastern Pacific using vocal dialects*. PhD thesis, UBC, 2005.

[Zar29] Boris Zarnik. Zivot i rad ivana regena. *Priroda*, pages 1–7, 1929.

Chapter 10

Appendix A

10.1 Alberta Biodiversity Monitoring Institute

Another project that collects large amounts of recordings of animals is the Alberta Biodiversity Monitoring project (ABMI), a project setup by the Alberta Government to monitor the biodiversity of ecosystems across the province.

Every summer, researchers are sent to a grid of over 1700 sites across the province, where they visually make note of plant and animal species they encounter. At the same time they are doing this, they set up a stereo microphone and record 30 minutes of audio from the site. During the following fall and winter, a set of researchers are employed to listen to these recordings and annotate individual birds in the recording. These annotations are then used to determine the changes in biodiversity in the environment. This process of annotating birds is done using a simple audio visualization tool (typically Audacity [LBF06]), along with a combination of Excel spreadsheets, websites that contain examplars of birdsong of different species, and other web based and electronic resources.

A collaboration with the ABMI has been established, in which the system developed in this thesis is being used to annotate the recordings for the ABMI. Considerable work has already been done to modify the interface to better suit the needs of the ABMI project, and work on this project is ongoing. It has been encouraging to note that the work done already on the web interface for the Orchive has been directly useful to this project, and that the changes to the interface have been easy to implement in the framework of Orchive Version 2.0.

In this chapter we will then provide a brief introduction to the analysis of vocal-

izations of the taxonomic class Aves, also known as birds. The mechanisms of sound production by birds is extremely diverse, and is an extremely large literature devoted to the study of the vocalizations of birds, so in this work we will only attempt to provide a general overview of the literature in this field. One of the primary motivations in studying large bioacoustic databases is for the monitoring of the biodiversity of ecosystems, and we will provide a detailed overview of the work done in the use of bioacoustics to study changes in biodiversity.

We will also provide a brief introduction to the field of Acoustic Ecology, and will show how the research into this field has informed this work and how it relates to studies of biodiversity.

10.1.1 Birds and Biodiversity

In the natural habitat, the sounds of animals can be used to determine how many animals are in a location and what species these animals are, and can help scientists measure the biodiversity of a location, and how this biodiversity changes over time. There is recent interest in this field of study [WTP+10] as a method for determining the biodiversity changes in regions over time. The authors then say that an automated system for detecting insects would also be useful from a commercial point of view,

The paper then goes on to say that in laboratory settings, it would also be advantageous to have an automated system that could segment and classify recordings. Currently this task often requires researchers to annotate hundreds of recordings by hand, which is a very time intensive task. As a scientist who has done many hours of hand annotating of large sound archives, I can attest to the difficulty and time-consuming nature of this task.

The authors then say that there are many problems with current methods of detecting, segmenting and classifying the sounds and vocalizations of animals. Current methods require researchers to manually annotate recordings and to generate a large corpus of data to be used by classification algorithms. These algorithms often have a large number of tunable parameters, some of which can be automatically determined from the large amount of training data, and some which have to be manually explored by the researchers. In addition, many of these algorithms are computationally expensive and cannot be deployed in the field. I have found these facts to be true in my own research, and I appreciate the fact that the authors spend time discussing the problems faced by researchers studying bioacoustics.

The use of bioacoustics for the study of species in the Amazon is presented in [Rie93], in which a system for doing rapid surveys of species in forested areas are presented. Challenges of doing surveys of species via visual surveys include the complexity of tree architecture, the inaccessibility of species, rarity, excellent camouflage, and cryptic and noctural lifestyles. In addition, the visual identification of species is labour intensive, costly and is in increasingly short supply. Many of these species can be heard, but cannot be seen, and this paper presents the challenges and advantages of using sound to identify these species. The challenges include the lack of resources of audio recording exemplars of these species, however mentions that there are many new projects devoted to the collection of sound samples from species. The advantages include the fact that many of these species have evolved so that their sounds are distinct from each other, and are thus amenable to study from acoustial fingerprinting. This paper is of relevance to the current work as it shows that the use of bioacoustics for monitoring diversity of species can be a useful approach.

An approach for doing a rapid survey of biodiversity is presented in [SPHD08]. In this work, the bioacoustic signatures of individual species are not identified, but rather an approximation of the biodiversity of a region is calcuated using an acoustic entropy index, in which the temporal entropy of a Hilbert-Huang transform of the sound is used to estimate the number of species in a region. From actual recordings from Borneo they find that the acoustic entropy has a logarithmic relationship to the number of species in a region. In addition, they find that in regions where environmental impacts have occurred, that spectral profiles show a higher dispersion of peaks of amplitude along the frequency axis. This paper is of some relevance to our work, however in this thesis we are primarily interested in the actual identification of individuals and species from the sounds they make.

An excellent review of the field of using automated sound recording and analysies for doing surveys of birds was done by Brandes in 2008 [SB08]. In this paper, two broad categories of automation are examined, the first being the automated recording of bird sounds, and the second being the automated analysis of these sounds. Different types of hardware recording devices are discussed, as well as different types of microphones suitable for this task. Of more direct relevance to this thesis is the section which discusses the automated analysis of bird songs, and divides bird vocalizations into 5 distinct categories, which are constant frequency, frequency modulated whistles, broadband pulses, broadband pluses with varying frequency components and segments with strong harmonics. They make note that different audio feature extrac-

tion and machine learning algorithm pairs are necessary for obtaining features from these different kinds of vocalizations, for example, for pulse to pulse duration features are useful and neural networks have been shown to be useful and for sounds with frequency modulated whistles or segments with strong harmonics, cepstral coefficients combined with dynamic time warping and hidden markov models have shown their effectiveness. This paper relates to this thesis in that we have developed a system to allow for the many hours of recordings captured by recording devices to be viewed and allows for a variety of audio feature extraction and machine learning algorithms to be applied to audio.

Three traditional methods of measuring bird biodiversity are stop counts, in which an observer travels a specific distance and stops to record all instances of birds, point counts, [BWK+10] and mapping of breeding territories. A recent paper [FBC08] describes how the use of bioacoustic approaches have assisted these traditional methods. Important ways that automated recordings can help is that the recording can be done in the absence of an observer, which can be of benefit in ecologically sensitive areas. In addition, noctural and low vocal activity can be studied with the use of long duration recordings. This paper describes several actual bioacoustic recording studies and talks about the advantages of this method. This paper is of direct relevance to the current work in that it shows how bioacoustic recordings can directly benefit studies in biodiversity.

In this same conference, a paper by Bardeli et. al [BWC08] describes the use of automated algorithms to detect bird songs in the results described above. In it, the song of the caffinch is detected using a three step procedure, in which first all end segments of a song are detected, then element repetition frequencies in a given time window before the song end are estimated, and finaly each combination of end sequence with repeated segments that are in a range consistent with caffinch sounds are reported. Dynamic time warping is used to detect the end segments, and an autocorrelation based method is used for the estimation of repeat frequencies. In the system built for this thesis, we allow for both the use of Dynamic Time Warping and Autocorrelation based methods, and would support the studies carried out in this paper.

In [BWK+10] the authors describe a system for performing censuses of bird populations using automated methods. Their method uses DSP based techniques that take advantage of both frequency content and temporal characteristics of bird songs, and describe filter chains that were optimized to detect two bird species, the Eastern

bittern and Sevi's warbler. Both of these methods are finely tuned to the individual characteristics of these two songs, for example, the song of the Eastern Bittern has a dominant frequency of 150Hz which is repeated in a specific temporal pattern. They perform frequency analysis of the audio and use a peak picking algorithm at 150Hz to detect candidate events, and then use an autocorrelation based method to lower the number of false positives. Although they show good results for these two species in isolation, they admit that scaling this method up to cases with multiple bird species would be difficult, and that the presence of noise in recordings is also a challenging for their method and could introduce false positives.

The field of Acoustic Ecology studies how living beings and the environment interact through sound [Wri00]. It was founded in the 1960's by Barry Truax [Tru01] and primarily involves the recording of soundscapes of environments sometimes combined with the creation of artistic pieces [Wes02] that include these soundscapes. These environments often include the sounds of animals and are often recorded in natural environments from wildernesses [Smi04] around the world [Fel94]. In the literature, there is often an overlap of the term acoustic ecology when talking about animals [Now05] [KM96], and although this is a distinct field from the acoustic ecology of Truax, there is considerable overlap in the methodology and the practice of deep listening of these two fields [Red98].

10.1.2 ABMI

The ABMI is a non-profit organization that collects, analyzes and reports on the status of biodiversity in the province of Alberta. It is one of the largest and most systematic such biodiversity projects in the world, and is widely regarded as a leader in biodiversity monitoring. It has the goal of refining, analyzing, and synthesizing raw data make it more meaningful for politicians, policy makers and managers to use.

The ABMI surveys 1656 evenly spaced monitoring sites across Alberta, and visits 330 sites each year, and surveys the entire province every 5 years. There are 80,000 species of wildlife in Alberta. About 2000 (¡ 2%) of these species are vertebrates, and vascular and non-vascular plants, and it is these 2000 species that are monitored by the ABMI. During these surveys, a team of two researchers make detailed observations of the site, collect samples for later identification, and of most relevance to this project, make high quality audio recordings of the site, which are later analyzed by highly trained researchers in bird identification.

It is important to measure biodiversity because in order for policy makers to make informed decisions when it comes to areas of sustainable resource management, it is necessary to first measure and quantify the current state of biodiversity, and to measure the changes in biodiversity that happen during projects. In order to do this, there is a need for a solid evaluation method for biodiversity that is objective and rigorous to evaluate changes in biodiversity. The ABMI monitors changes in highly relevant species and uses a pyramid framework for aggregating data 10.1. This pyramid has at its base the primary data, this is synthesized in progressively higher levels of abstration to the state of individual species, specific guild-level indices, general guild-level indices, to taxonomic indices and finally to a single index. Every year, this single index is further summarized into an annual report that describes broad changes in biodiversity in the province of Alberta. The communication of results is of primary importance in this project, and all results are freely available to all interested parties.

The ABMI was designed to increase efficiency by using a cost-shared operational model which reduces duplication of effort and it benefits many sectors of the economy including forest sector, energy sector, agricultural sector, the public, the government and the scientific community

Biological diversity (biodiversity) can be defined as the variability among living organisms from all sources including terrestrial and aquatic ecosystems. At a broader level, this includes the the ecological complexes of which they are part, and the diversity within species, between species, and of entire ecosystems. Biodiversity is a shared responsibility of at least 17 different agencies in Alberta, and the ABMI functions as a central repository of data on biodiversity

Cumulative-effects monitoring approach that is targeted at detecting the ecological effects of a diverse set of environmental stresses on broad suites of indicators [MZSM04]. Cumulative-effects monitoring exposes correlative relationships between stressors in a system and the many indicators that are monitored [TSH94]

ABMI assesses the performance toward management objectives such as "regional sustainability" or "ecological integrity" [MNS+99]. This monitoring approach remains relevant over long timeframes as new human activities and environmental stresses are introduced to landscapes [WN04].

The province of Alberta is 661,000 km2 and makes up 6.6% of Canada. Alberta has 6 major Natural Regions (Natural Regions Committee 2006). The Rocky Mountain Natural Region makes up 7% of Alberta and is dominated by rock, icefields,

and coniferous forest. The Foothills Natural Region makes up 10% of Alberta and is dominated by rolling topography with deciduous, mixedwood, and coniferous forests. The Grassland Natural Region makes up 15% of Alberta and is dominated by grasses and shrublands. The Parkland Natural Region makes up 9% of Alberta and is dominated by deciduous forests and willow shrublands. The Boreal Forest Natural Region makes up 58% of Alberta and is dominated by deciduous, mixedwood, and coniferous forests. Finally, the Canadian Shield Natural Region makes up 1% of Alberta and is dominated by exposed bedrock, and open deciduous and coniferous forests.

The ABMI does a systematic and unbiased sampling scheme to survey biota and habitats. In addition, for a given sampling effort, creating a series of panels and surveying one panel each year, achieves higher statistical power than having fewer locations and surveying all locations each year [UPL98] [UK99].

The ABMI uses many raw data sources for its reports, including field notes of physical site characteristics, tree cores, plant samples and photographs, but the data source of interest in this thesis are the audio recordings of birds at each site.

These bird recordings follow a detailed protocol which each team is trained to follow. For each site, there are 9 different recordings that are taken. These recordings are done in a grid pattern, starting at the site location, with each subsequent recording done 300 meters from the last, in a proscribed fashion as shown in Figure 10.2. The bird recordings are begun approximately 30 minutes before sunrise, and during the recording, the person doing the recording is instructed to listen on headphones during the entire recording to ensure it is recorded. The second member of the team is instructed to remain silent during the recording to not disturb the birds and to avoid noise that could be heard on the recording. The recordings are done with an advanced omnidirectional microphone well suited to recording birds, and will be described in detail below. The recording levels are standardized so recordings are comparable and to avoid potential for poor recordings, the recording level is set based on the value for the particular model of microphone being used, this setting is used consistently among sites regardless of wind or environmental conditions.

All recordings are made at 320 kbps or greater sampling rate, which means that the maximum frequency that can be recorded is 160kHz, which captures most of the main frequencies and harmonics of birds.

At the beginning of each recording the person doing the recording is asked to state their name, crew number, date, site number, point count station number and the start time, this is followed by a standard tone that is at least 5 seconds in duration,

which is used to standardize recording intensity. During the 10 minutes in which the recording is being made, the recordist scans the surrounding area using binoculars, and for all birds observed they state the species identity, number of individuals, and distance they are from the microphone so that this information is also recorded.

To ensure data is not lost, the 9 bird recording files are digitally copied to a laptop computer or portable hard drive at the end of each day, and all recordings are saved on the recorder as well as on the laptop for backup in the field. At the end of each field shift, bird recordings are transferred to the lab. To maximize detections of birds surveyed are conducted between May 25 and June 22.

The omnidirectional microphone that is used in these recordings is an advanced CZM microphone, which can be seen in Figure 10.3. This microphone has a 360 degree radial pattern and provides a forward sound pressure gain comparable to a 24-inch parabolic microphone. The CZM microphone is a radially directional acoustical microphone which is designed to simultaneously receive both direct and reflected sound waves at its boundary. These sound waves are then compressed as they move through the flat plane of the microphone enclosure toward the microphone diaphragm. It increases the signal-to-noise ratio by 14 db, by using a wave guide design. This wave guide design is shown in Figures 10.4 10.5.

These recordings are then analyzed by a highly trained team of experts in bioacoustics and bird identification. The tools that are used by these experts are standard programs which include Audacity [MD02] for listening to and viewing the sound, Microsoft Excel for entering data, and a variety of web browser windows that have example clips of bird vocalizations, common birds in the study area, and other types of information about these birds. The process for listening to and identifying bird vocalizations using these tools is cumbersome, and recently we were contacted by the head of this team to help design custom software to help with this process. We subsequently obtained funding from the ABMI to develop a tool to help with this process.

In this thesis, we describe a software system that we designed in collaboration with this expert to help him and his team perform the task of identifying bird species in these recordings. This tool incorporates a considerable amount of the code and tools designed for the Orchive interface, and the additions made this tool in the course of the ABMI project has made it more flexible and capable of dealing with different forms of audio data. We present results from a iterative development process in which we co-created a tool with the experts from the ABMI that allowed them to listen

to, view and annotate this large bioacoustic archive in a collaborative manner. In addition, preliminary results of automated identification of birds from these recordings is presented.

This development process highlights the importance of developing highly interactive tools that empower experts using technology designed with their workflow and needs in mind. While it is possible to use standard tools to do this, the use of customized tools can greatly speed up the process of species identification.

Figure 10.1: A diagram showing the progression of data from the level where it is collected at the bottom up through derived knowledge and finally leading to public policy at the top.

In the case of the ABMI, these are the calls and songs of the birds that could be extant in Alberta in the period of the study session. For the calls of different species, the short time of each call, and the similarity between calls of different species makes this a challenging classification task. The songs of birds have more variation and are of longer duration, and are easier to classify. Birds that are taxonimally related to each other make similar sounds, for example the Swainsons thrush and the Hermit thrush have very similar songs, but differ in the raspiness of the first note of their song, which could be an identifying feature for machine learning systems to identify.

Figure 10.2: A diagram showing the proscribed manner at which the 9 different data points are recorded for each site in the ABMI data collection protocol.

10.1.3 Bird song identification

The data analysis protocol for the ABMI data is considerably different from that used to classify vocalizations for the Orchive. At each of the 1656 sample sites, a series of 9 different recordings in a grid with 300 meters between each point are collected. At each of these points, a duration of 10 minutes of audio are collected. After the field season, these recordings are analyzed by an expert in bird song according to a specific protocol. First, the recording is split into sections of three minutes each, and for the first of these, the first singing instance of an individual bird is noted, with a time resolution of 10 seconds. The presence or absence of the same individual is marked the second and third recordings, and if new individuals are present, the first singing times of these individuals are also recorded. It should be noted that these birds sing multiple times in each recording, but only the first instance of singing is recorded.

In order to test the effectiveness of audio feature extraction and machine learning on this dataset, a small dataset consisting of 67 clips with a total length of 86 seconds from a series of 5 recordings was obtained. These clips were from a total of 9 different species.

Figure 10.3: An image of the CZM microphone, used to efficiently collect a stereophonic field of sound but with the ability to amplify distant sounds.

Figure 10.4: A schematic view of the CZM microphone, used by the ABMI to efficently capture the sounds of birds in field recordings.

We then extracted audio features from these sections of audio using the Marsyas [Tza08] audio feature extraction framework. Marsyas has a wide variety of audio features that it can calculate, including MFCCs, number of zero crossings per window and various high level descriptions of the spectrum including the centroid (center of mass of the spectrum), rolloff (the frequency for which the sum of magnitudes of its lower frequencies are equal to percentage of the sum of magnitudes of its higher frequencies) and the flux (the norm of the difference vector between two successive magnitue/power spectra).

We then used these audio features as input to a number of common machine learning algorithms in the Weka [WF05] machine learning package. The first of these was the C4.5 decision tree algorithm, and at each step creates a new branching point based on the highest gain in information entropy that can be chosen at that point. The second was the Naive Bayes classifier which uses Bayes theorem, but considers each

Figure 10.5: A top down view of the CZM microphone, showing the way that sound is amplified simply with the physical design of the microphone.

feature to be independent from each other feature. The Random Forest algorithm uses decision trees like the C4.5 algorithm, but instead constructs a large number of decision trees and chooses the classification that has the highest number of decision trees that have chosen that classification. The Multilayer Pereptron is a variant of an Artificial Neural Network, and contains a number of hidden layers, each of which is trained using backpropogation. Finally, we also tried a Support Vector Machine (SVM) algorithm, which finds a hyperplane that best separates the high dimension input feature vectors. In Weka, the SVM is optimized using the Sequential Minimal Optimization technique, which separates the SVM problem into a series of sub-problems that are then solved using analytical methods.

The results of these machine learning problems on this small dataset are shown in Table 10.1, and from these results, it can be seen that SVM produced the best results. In addition, a confusion matrix for the SVM classifier is shown in Table 10.2. From this table, one can see that species that have songs that sound similar are misclassified, for example, the Tennesse Warbler is classified correctly 4 times, and is misclassified as a Palm Warbler 2 times.

One of the reasons of low accuracy of these classifiers is that these recordings do not consist of isolated vocalizations, and that birds frequently vocalize at the same time. Although the accuracy of these classifiers are low, because each bird vocalizes many times throughout the recording, even if a single vocalization is misclassified, it is possible that subsequent vocalizations will not contain cross-contamination from other species, and could be correctly classified. In order to improve these results, we are actively at work on extending our framework to handle Multiple Label machine learning classifiers which can handle multiple species vocalizing at the same time.

Classifier	Accuracy (% Correct)
C4.5 Tree	43.2
Naive Bayes	49.3
Random Forest	52.2
Multilayer Perceptron	61.2
SVM SMO	68.7

Table 10.1: Results from using different machine learning classifier algorithms on the problem of bird song classification. The accuracy of each classifier in percent correct classification is shown.

	ALFL	CHSP	HETH	OCWA	PAWA	SWTH	TEWA	WTSP	YBFL
ALFL	8	0	0	0	0	0	0	0	0
CHSP	0	8	0	0	0	0	0	1	0
HETH	0	2	7	0	0	1	0	0	0
OCWA	0	2	0	2	2	0	0	1	0
PAWA	0	1	1	0	7	0	0	0	1
SWTH	0	0	1	0	1	3	0	0	0
TEWA	0	0	0	0	2	0	4	0	0
WTSP	0	1	1	1	1	0	0	4	0
YBFL	0	0	0	0	1	0	0	0	3

Table 10.2: A confusion matrix for classifing bird song using a Support Vector Machine classifier. The vertical axis describes the ground truth label for each species, and the horizontal axis describes what label each clip was classified as.