DATA MINING IN LARGE AUDIO COLLECTIONS OF DOLPHIN SIGNALS

A Thesis
Presented to
The Academic Faculty

by

Daniel Kyu Hwa Kohlsdorf

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing, College of Computing

Georgia Institute of Technology
August 2015

DATA MINING IN LARGE AUDIO COLLECTIONS OF DOLPHIN SIGNALS

Approved by:

Professor Thad Starner, Advisor
School of Interactive Computing,
College of Computing
Georgia Institute of Technology

Doctor Denise Herzing
Wild Dolphin Project

Professor Irfan Essa
College of Computing
Georgia Institute of Technology

Professor Michael Beetz
Computer Science
University Bremen

Professor Charles Isbell
College of Computing
Georgia Institute of Technology

Date Approved: July 2015

ACKNOWLEDGEMENTS

The presented dissertation is the result of a process of an intellectually stimulating environment and several collaborators that helped to guide me through the research process. First I'd like to thank my advisor Thad Starner. Thad's advice, guidance and patience are unmatched. He taught me much by providing challenging problems and pointing me to interesting knowledge. His care went beyond my research, and he provided career guidance and was willing to discuss any idea brought to him. His approachability led to the most stimulating discussions, and he created one of the most wonderful research laboratories in the world. I could not have picked a better advisor. Irfan Essa always helped to maintain the big picture and asked the right questions about my research goals. He also made sure that I met other students from other labs with related problems. I made great friends that way. He provided lots of the necessary focus to complete this work. The collaboration with Denise Herzing is the main inspiration for this work. The ideas and challenges helped greatly to design the system presented in the thesis. Furthermore, she patiently shared her field experience and biology domain knowledge. I could not have finished this work without her constant interaction. Charles Isbell provided great discussions by questioning the relation of my work to the two fields this dissertation spans: behavior analysis / biology and machine learning. Finally, I want to thank Michael Beetz who provided a different view on my machine learning efforts, and the collaboration with him led to interesting ways of incorporating domain knowledge into my system. Altogether I could not have picked a better committee. Furthermore, I'd like to thank all members and former members of the Contextual Computing Group who created a great and open working environment. Especially, I'd like to thank David Minnen and James

Clawson. While David's work inspired lots of the algorithms in this thesis, he also took the time to discuss his and my results, and he provided valuable input for this work. James Clawson provided great feedback and advice for the qualifying exam, my proposal and my research in general. He made most of my presentations, talks and documents better. He also helped me arrive in Atlanta in my first year at Georgia Tech. Along similar lines, Helene Brashear was available for a lot of consultation, and she attended and fixed most of my talks. Besides my committee I got lots of help from others. Tavenner Hall has proof-read not just this thesis but also all texts I produced since I arrived at Tech and provided most of the moral support during my PhD studies. A special thank you goes also to Celeste Mason who kept me sane through the later part of the thesis writing and kept my motivation throughout the dolphin project. Finally, I would like to thank my parents, Eva Rack-Kohlsdorf and Wolfgang Kohlsdorf. My father Wolfgang sparked my interest at an early age and might be the reason I decided to study computer science. Finally, my mother Eva supported me throughout my life in all things I have done. I owe most of my education and success to her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The study of dolphin cognition involves intensive research of animal vocalizations recorded in the field. In this thesis I address the automated analysis of audible dolphin communication. I propose a system called the signal imager that automatically discovers patterns in dolphin signals. These patterns are invariant to frequency shifts and time warping transformations. The discovery algorithm is based on feature learning and unsupervised time series segmentation using hidden Markov models. Researchers can inspect the patterns visually and interactively run comparative statistics between the distribution of dolphin signals in different behavioral contexts. The required statistics for the comparison describe dolphin communication as a combination of the following models: a bag-of-words model, an n-gram model and an algorithm to learn a set of regular expressions. Furthermore, the system can use the patterns to tag dolphin signals automatically with behavior annotations. My results indicate that the signal imager provides meaningful patterns to the marine biologist and that the comparative statistics are aligned with the biologists' domain knowledge.

CHAPTER I

INTRODUCTION

Dolphin cognition and communication research is a significant sub-field of marine mammalogy. Communication signals of animal groups can give valuable insight into their social structure. One of the goals in dolphin cognition research is the association of social cues during group behavior with audible signaling by correlating video with audio recordings. Therefore, researchers collect large multimedia databases in the field containing long-term behavioral observations.

However, animal communication research suffers from the slow speed of manual data analysis. Often researchers search and annotate audio and video material using manual measurements. These measurements are subjective and not formally defined. Finding patterns of communication that relate to observable behavior without metrics for comparison is a tedious process. The process, from data collection to publication, can take several years or even decades. In the following dissertation, I propose an interactive data mining system that supports marine mammalogists as they search multimedia databases more efficiently and inspect their data using statistical testing. I will give a short overview of current behavior research in dolphin communication, describe the key challenges for automatic dolphin communication mining and finally provide my thesis statement and my contributions.

1.1 Behavior Analysis of Atlantic Spotted Dolphins

Marine biologists can now collect large multimedia databases of wild dolphin behavior in their natural habitat using cameras and microphones. The Wild Dolphin Project [16], has collected over 29 years of data of wild Atlantic spotted dolphins (*Stenella frontalis*) in the Bahamas for 100 field days every summer. As the biologist

team observes the dolphins, they use multiple underwater cameras and hydrophones to capture visual and audio data. Each encounter with dolphins is about 10 minutes to an hour long. As the researchers observe dolphins underwater, their video data captures a wide variety of specific behaviors and social contexts such as nursing and aggression.

This database is a rich resource for animal behavior analysis. The collected data can give insight into the social structure and communication patterns of a species. An example use of such a database is a study of the dolphins' signature whistles that shows how dolphins can use whistles as names for themselves and others to maintain social bonds [21]. To establish this evidence, over 200 hours of acoustic recordings of temporarily caught-and-released, wild bottlenose dolphins had to be annotated by one observer and segmented manually. Often the whistles must be traced manually as an intermediate step in order to allow researchers to establish a high intra-rater reliability [20]. This form of analysis is tedious and a time-consuming undertaking. Conversations with behavior researchers reveals that every hour of animal behavior requires 10 hours of manual analysis. Due to this large delay, most datasets are partially explored, and inspecting the complete picture of animal behavior across multiple contexts is not possible.

1.2 Search in Audible Dolphin Communication

My goal was to design a system that can automatically find patterns in audible dolphin communication and correlate these patterns with different dolphin behaviors and contexts. One of the biggest challenges is to define conditions under which two recordings of audible dolphin signals are similar. Biologists studying animal behavior determine similarity of audible communication by manual frequency measurements and visually inspecting spectrographic displays. Signals in these categories have common acoustic characteristics. However, computer-aided approaches to communication

mining are widely unexplored, and the fundamental units of communication remain unknown. Without tools that support the analysis of dolphin communication, every hour of audio recordings requires 10 hours of manual analysis. Researchers collect large amounts of field recordings every year. The slow analysis techniques lead to large subsets of the data remaining unexplored. For example, in a study analyzing the communication patterns of Gunnison's prairie dogs (*Cynomys gunnisoni*), researchers found that the animals code information about the type of threat in their alarm calls. However, the analysis process from collection to publication took several decades [46].

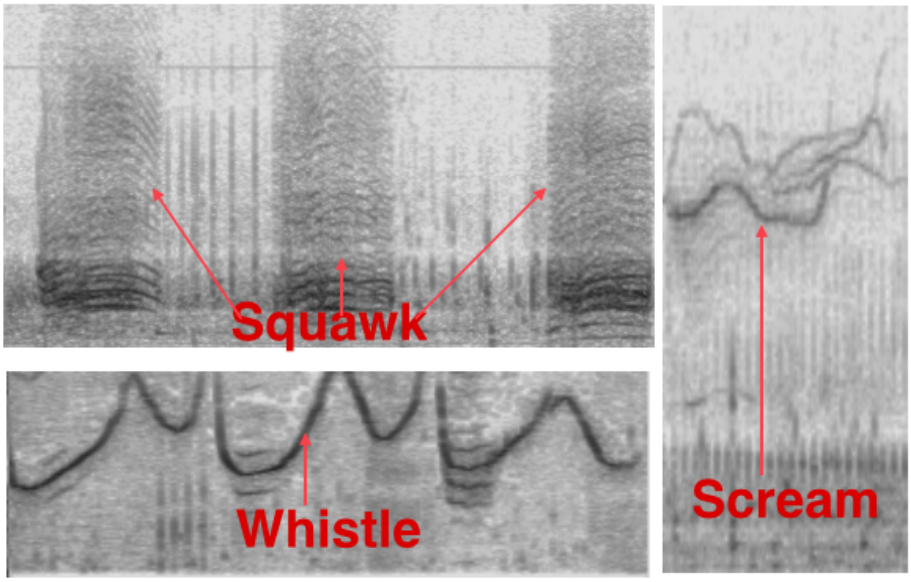


Figure 1: Different audible dolphin communication signals.

Using visual inspection and manual measurements, biologists have created various categorizations for audible dolphin signals. For example, two common categories are dolphin whistles and dolphin burst pulses. A dolphin whistle is thought of as a single oscillator changing frequency over time. In a spectrogram a whistle appears as a single line, where each point on the line represents the frequency of the oscillator at a specific time. A burst-pulse signal is a dense series of loud clicks. In the spectrogram, a burst-pulse series shows as multiple parallel lines. The spacing between the lines

represents the repetition rate of the clicks.

As one can see in Figure 1, the dolphin whistle (bottom left) is visually dissimilar from a burst-pulse sound (top left, labeled "Squawk"). It is interesting that the sounds in these categories are not only dissimilar in their appearance but also differ in their usage. Behavior researchers found that specific dolphin whistles called "signature whistles" are used by dolphins to identify each other, while a specific type of burst pulse called a "synchronized squawk" is used by dolphins during aggressive behavior. Finding patterns and subcategories in these defined categorizations might give a more detailed view supporting research on dolphin communication and its relation to behavior.

The pattern discovery algorithms are invariant under two transformations that can be found in dolphin communication "frequency shift" and "time warping." In other words, two dolphin signals should be similar given a distance function even when shifted in frequency or warped in time. A signal shifted in frequency will appear with the same shape, but all points are translated by the same amount upwards or downwards in frequency. An example of a frequency shift in human communication is speaking in a lower register. All the words are the same, but they are uttered in a lower frequency space. A time warped signal is stretched or shrunk in time. The example in human speech is to utter a word faster or slower.

In both cases, the words maintain the same meaning- their spectral shape will remain uniquely distinguishable- allowing other humans the ability to recognize the word under these transformations [1]. An example of these transformations for a dolphin whistle is shown in Figure 2, left.

In Chapter III will describe my approach to finding subcategories or patterns using

[1]Of course words become unrecognizable if the speed of the word is artificially high or the word is artificially produced outside the range of human hearing. However, in day-to-day use, humans are capable of dealing with these transformations. In our datasets I am not expecting drastic transformations since the signals are generated by a biological system. In other words, the communication between animals relies on the signals to be recognizable and distinguishable.

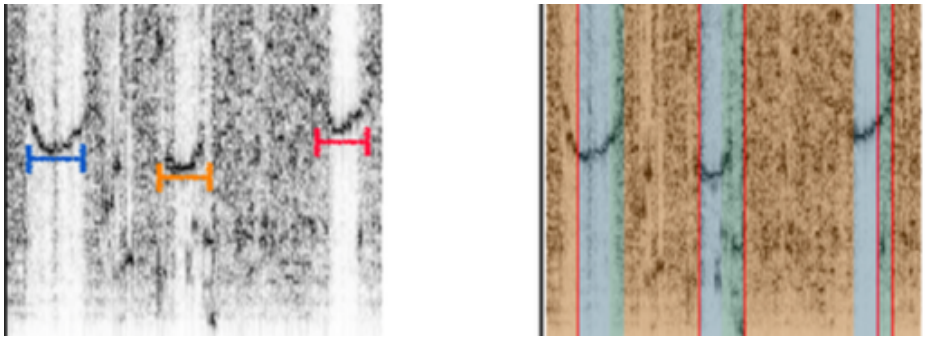


Figure 2: Left: An example where three patterns similar in shape vary in time and frequency. Right: A segmentation result from my algorithms.

similarity scores invariant to frequency shift and time warp. Computing statistical communication models of sequences of patterns as they occur in the continuous underwater recordings can then help to analyze the audible communication in different contexts as described in Chapter IV.

1.3 Annotation of Dolphin Communication

My system is designed to analyze audible communication in different contexts. The system uses discovered patterns to predict which annotations are appropriate. In this section, I describe the annotations currently made by biologists. The goal is to use these annotations later to train a machine learning algorithm.

After collecting audio and video data of wild dolphin behavior in the field, the biologists perform a quick retrospective analysis. By watching the video feed the experts search for typical dolphin behavior such as a dolphin slapping another dolphin with it's tail or multiple dolphins swimming head to head (see Figure 19). These actions are often associated in the marine biology community with higher level behavior contexts such as play behavior or aggressive behavior. Furthermore, the dolphin researchers try to identify dolphins by visual features on their bodies. Examples of visual clues identifying a dolphin include spots on the dolphin or scars. Other observations from the video include objects or relations. For example, the presence of a dolphin toy such as sargassum (a seaweed) or the presence of a calf are possible observations that

are not a visually identified behavior or a dolphin ID. These annotations are recorded with a time stamp on the video. The researchers also annotate some of the audio's spectrogram. Common annotations include the type of a dolphin signal. Example annotations include whistles, signature whistles and echolocation. All the annotations from a dataset collected in 2012 are shown in Table 1. Each label represents a short form for an annotation such as a dolphin ID or a behavior. If a label is described as a "Visual ID", it refers to a dolphin identified from video. Labels described as "Spectrogram Inspection" refer to signal types as identified by manual inspection in the audio's spectrogram. If a label is referred to as "Visual Behavior", it describes a behavior as observed in the video and if it is described as "Visual Observation" it represents other observations made from the video.

An interesting task is to try to predict these annotations based on the patterns found in the acoustic dataset. Predicting annotations that are made from observations of the spectrogram seems straightforward. However, if a system could use acoustic features to predict annotations made based on visual cues, it would suggest a meaningful association of the acoustic and visual contexts. For example, imagine a video of a dolphin. The researchers are able to identify this dolphin as "Bishu." Suppose the system discovers patterns in the audio feed and predicts correctly that "Bishu" is in the frame. Now the researchers can inspect the patterns in the spectrogram and might conclude that a whistle in the audio stream is in fact "Bishu's" signature whistle as used by the dolphins as a call sign. Such successfull predictions could also be indicative of the performance of the pattern discovery. That is, meaningful patterns could be predictive of at least some of the annotations.

1.4 Thesis Statement

The need to automate the search for patterns in audible dolphin communication under various transformations and to find statistical models of communication in

different behavioral contexts leads to my following thesis statement: *"I hypothesize that feature learning and automatic segmentation of audible dolphin communication along with statistical communication models can provide valuable insight into dolphin behavior that allows marine biologists to perform retrospective analysis as well as scientific hypothesis generation and testing."* Modern feature learning and segmentation approaches provide a framework in which I can achieve pattern discovery invariant to frequency shift and time warping transformations. Automatically finding these patterns in dolphin communication will reduce the amount of time needed for analysis and will lead to less subjective measurements when comparing signals. Furthermore, finding statistical models for the occurrence of these patterns in different behavioral contexts can provide a framework that can be used to generate, test and evaluate novel hypotheses about dolphin communication. In order to implement and test such a system, I provide the following contributions:

1. A feature learning algorithm for dolphin signals that results in frequency invariant features. The features allow the algorithm to distinguish between dolphin signals and other underwater noise sources and allow it to distinguish between dolphin patterns using hidden Markov models.

2. A warp-invariant pattern discovery algorithm that discovers subcategories in dolphin communication sequences.

3. A statistical model for dolphin communication based on n-grams and regular expressions that can be used to perform comparative statistics about different behavioral contexts and to tag unannotated communication sequences.

4. A user interface to the system, providing biologists with a tool called the signal imager that allows to run discovery experiments in dolphin communication.

Table 1: A list of all the annotations found in the 2012 dataset.

Label	Description	Num.
Whistle	Spectrogram Inspection	41
Littleprawn	Visual ID	9
Bamboo	Visual ID	9
Nuzzle	Visual ID	9
Littlegash	Visual ID	8
Bishu	Visual ID	8
Nautilus	Visual Behavior	6
Play	Visual Behavior	6
Head2Head	Visual Behavior	6
Ginger	Visual ID	5
Gelato	Visual ID	5
Signature Whistle	Spectrogram Inspection	4
OpenMouth	Visual Behavior	4
Echolocation	Spectrogram Inspection	4
Malachite	Visual ID	4
Mugsy	Visual ID	3
Cobalt	Visual ID	3
Sync	Spectrogram Inspection	3
TailSlap	Visual Behavior	3
Naia	Visual ID	2
Nematocyst	Visual ID	2
Val	Visual ID	2
Sargassum	Visual Observation	2
Chase	Visual Behavior	1
Calve	Visual Observation	1
Fecal	Visual Observation	1
Fish	Visual Observation	1
Nautilus	Visual ID	1
Flexion	Visual Behavior	1
Venus	Visual ID	1
Fused	Visual ID	1
Discipline	Visual Behavior	1

CHAPTER II

RELATED WORK

In the following chapter, I will describe previous work related to my research. First I will introduce current research in animal communication. This section will discuss how manual analysis is performed and describe research in automated analysis in dolphin communication. I will then describe previous research in feature learning. The results will inform the design of features tuned for dolphin signals. The third section will describe previous research in motif discovery that is the main inspiration for my system. Lastly, I will describe multiple natural language processing approaches and their relation to the statistical analysis of dolphin communication.

2.1 Animal Communication Analysis

A common solution to the manual analysis of animal communication is to use interactive computer programs such as Cornell Lab of Ornithology's Raven and Noldus Information Technology's Observer. Cornell's Raven [10] is an audio processing program. Researchers can visually inspect their audio recordings and annotate animal signals in the spectrogram interactively. Raven includes signal detectors that can segment an audio file into regions of noise and regions where animal signals are present. However, the included signal detectors are basic algorithms such as band-limited energy detection. In my opinion, the disadvantage is that noise in the same frequency band as animal communication can lead to false positives during detection. For this reason, analyzing noisy samples can be difficult. Furthermore, the program includes simple techniques to compare signals such as correlation between spectrograms. However, the correlation coefficient does not account for transformations such as frequency shifts and time warping effects. While Raven requires excessive manual effort from the

user, the signal imager automates most of the work by discovering patterns automatically. Instead of searching for patterns manually in the spectrogram and annotating patterns by visual inspection, my program automates most of the work.

Noldus Observer enables researchers to code behavior with annotations in video and audio. Annotations or behavior codings of audio and video files can be extracted from Observer and used in my program to train automatic annotation algorithms and to perform statistical analysis. While the program offers easy behavior coding, it does not support any audio analysis capabilities.

In a survey on underwater acoustics processing methods, Lampert and O'Keefe [25] identify three main algorithm categories: image processing, neural networks, and statistical models. They evaluate several methods on a dolphin whistle detection task. Their experimental parameters include signal-to-noise ratios, noise variation, whistle shape variability, multiple whistles, between-whistle proximity/crossing, initial/endpoints of whistles, and computational resources used. They conclude that hidden Markov models (HMMs) are currently the most prevalent, promising method in the research literature for use in cetacean vocalization spectrum analysis.

Kershenbaum et al. [20] measure the similarity between whistles using the dynamic time warping distance. Whistle extractions are performed manually using a custom user interface. Users manually follow the contour of the whistle in a spectrogram. However, this task can be performed automatically as shown by recent efforts of Baggenstoss and Kurth [2] who compare methods for detecting burst pulses in impulsive noise and Kohlsdorf et al. [22] who trace a dolphin whistle using a probabilistic pitch tracker. Other approaches to whistle extraction include a frame-based Bayesian approach [15] and a Kalman filtering approach [24]. Shapiro and Wang apply pitch detection designed for human telephone speech to whale vocalizations [42].

Dolphin signal clustering and classification often uses neural networks [11, 14]

or clustering based on hidden Markov models [1]. Both approaches filter the data first and use Mel-cepstral coefficients or other measurements from the spectrogram as features. My goal is to improve the efficiency and efficacy of analysis of clicks, whistles, and bursts [17]. We adopt an approach similar to the work done by Zakaria et al. on mining archives of mouse sound using symbolic representations [52]. In this work, known units of mouse vocalizations are retrieved under the generalized Histogram of Oriented Gradients (HOG) transform, and strings of these units are compared using hierarchical clustering under the Levenshtein distance.

My system combines several methods from the above categories. Image processing methods similar to convolutional neural networks are used as the basis for feature extraction [9].

I use feature learning on a dataset of several patches found around traced whistles and other dolphin signals. I use hierarchical clustering to identify units. Similar to Zakaria, I proceed to process the units as discrete strings. However, instead of using an alignment score (Levenshtein distance) I use the alignment to extract a set of regular expressions. In contrast to most dolphin communication research, our system is able to process a wide range of dolphin signals, not just whistles. In the following I will describe the related work leading to my system; in particular I describe the process of learning a better feature space for dolphin communication, discovering patterns under time warping transformations and statistical dolphin communication models.

2.2 Signal Processing

The spectrogram is the most prevalent means of analyzing dolphin communication. For this system, my goal is to learn a feature space that is tuned to dolphin communication only. Recently, a novel paradigm of signal processing called "self-taught learning" [39] has emerged in the machine learning community. The goal is to learn

a set of features from unlabeled data. The most prominent models for self-taught learning are probabilistic neural networks such as the restricted Boltzmann machine [18] and the convolutional restricted Boltzmann machine [28]. These models show promising results on visual object recognition and speech processing tasks. A similar method to learn a set of features is called sparse coding. In sparse coding, the goal is to learn a codebook of basis functions [27]. An unseen example is described as a weighted linear combination of these bases. The weight for each basis can be used as a feature.

A 2011 experiment with several computer vision tasks found that a k-means codebook can outperform single layer neural networks for feature learning [9]. Since neural networks have a large parameter space, my intuition is that the performance of neural network approaches is dependent on the amount of data ready for training. I will use k-means to learn a set of feature extractors tuned to dolphin communication. Since dolphin signals' appearances vary in the spectrogram, this method will help to convert audible communication into a feature space in which signals are easily comparable.

2.3 Motif Discovery

In order to find patterns in dolphin communication, I use a technique called motif discovery. First I learn a feature space in which dolphin communication is easy to compare. Then I discover common patterns in dolphin communication in this new feature space.

The bioinformatics community is very interested in discovering meaningful patterns and has developed algorithms to find patterns in symbolic sequences representing bases in a genome. Often, data mining approaches based on discretization are inspired by earlier work in biology. One prominent solution to finding patterns in multiple sequences is the multiple sequence alignment. An alignment arranges two sequences such that similar regions are identified. A multiple sequence alignment

arranges multiple sequences to find similarities across all files. This alignment can be achieved using Gibbs sampling [26] or a profile hidden Markov model [13]. Discovering so-called time series motifs can be achieved using distance comparisons [36] between sliding windows. The algorithm can find the two closest windows efficiently. However, there are several drawbacks to the approach. One is that the patterns have to be of equal length. Furthermore, in large datasets it might take too many distance comparisons to find all patterns.

A more efficient approach to pattern discovery is to discretize the time series first. A prominent algorithm for time series discretization is called symbolic aggregate approximation (SAX) [29]. SAX converts all time series into a symbolic string over a finite alphabet. The string resembles the shape of the time series. Increasing the size of the alphabet will result in a higher resolution of the discretization. The indexable symbolic aggregate approximation (iSAX) [43] allows the efficient search for time series in multiple resolutions. One efficient way of finding patterns is based on the iSAX representation. The idea is that similar time series will have the same symbolic representation. All sequences are converted into the symbolic representation and then inserted into a hash table. All sequences that end in the same bin are considered a pattern. By performing the process with iSAX, it is possible to find patterns at multiple resolutions [7]. The problem with iSAX is that the symbolic space is very large for high-dimensional time series, such as a spectrograms. Since dolphin communication is often analyzed in the spectrogram, the iSAX representation is not well suited for my use in this case.

Another efficient approach to pattern discovery in a symbolic space is based on random projections [6, 8, 32]. Minnen et al. [32] extract sliding windows from all time series in a dataset and convert each window into the SAX representation. Then the algorithm selects random positions and deletes these positions from all strings. This process is called a random projection. All similar strings are hashed again. The

strings that match most often under several random projections are considered to be a pattern. The advantages are that the sequences do not have to be compared at all positions and that the approach is very robust with respect to noise.

Park and Glass proposed to find patterns by developing a local alignment version of the dynamic time warping distance called segmental dynamic time warping [37]. It is used for automatic speaker segmentation. Aligned cluster analysis uses the dynamic time alignment kernel to cluster segments of a time series in a kernel k-means fashion [53]. Saria et al. [41] use a more general probabilistic graphical model to learn a deformable pattern model based on splines. Both approaches have the advantage that the patterns do not have to be equally long.

Smyth proposes using the Baum-Welch algorithm to train a mixture of hidden Markov models [47] to cluster time series. Gaussian mixture models combine multiple normal distributions into a joint probability distribution. In the same way, a mixture of hidden Markov models combines multiple hidden Markov models into a larger one. Each of the hidden Markov models represents a cluster. A time series belongs to the cluster with the hidden Markov model that returns the highest likelihood for that sequence. Minnen et al. estimate such a mixture by greedily adding the hidden Markov model that most increases the likelihood for the complete dataset [34, 33].

Several related works in the biology community already use dynamic time warping and hidden Markov models as models for dolphin signals; I decided to base my algorithms on the same models. Both algorithms account for the previously mentioned time warping effect. I use dynamic time warping to cluster patterns and use these clusters as an initialization for a mixture of hidden Markov models. The algorithm is inspired by Minnen et al.'s work [33] that shows that mixtures of hidden Markov models can discover patterns in complex domains such as human speech and activity recognition.

2.4 Communication Modeling

After I discover patterns, I want to model a dolphin communication sequence in terms of the pattern composition. I chose to use statistical models commonly used in systems that analyze human language. There are several approaches to modeling natural language, such as text. These models will later help me to annotate unseen dolphin communication sequences and to perform comparative statistics between communication in different behavioral contexts.

A common model is called a bag-of-words model [40]. In a bag-of-words approach, a document is described as a loose collection of words. In other words, the model does not consider the sentence structure. The only feature is the frequency of the words. For example, a document about biology might have a high frequency of words like "cell" or "DNA," while in a physics document words like "force" or "gravity" might be more frequent. When comparing documents under that model, one assumes that documents that are semantically similar will show similar words. The latent Dirichlet allocation (LDA) [4] is an extended model that represents documents as mixtures of topics. Similar to the bag-of-words model, a topic is a probability distribution of words. However, the document is described as a collection of topics.

Another model is called an n-gram model [40]. A n-gram model does not consider single words but local sequences of words. All the sequences are n words long. For example, all bi-grams for the sequence, "The fox jumps over the fence" are "The fox," "fox jumps," "jumps over," "over the," "the fence." In comparison to the bag-of-words model, n-grams model the local interaction of words. In our example, the n-gram model captures that the fox jumps. In a bag-of-words model that information is lost.

The final model I discuss is based on formal grammars. A formal grammar is a set of rules that is capable of generating and detecting sequences from a formal language. The complete description and theory behind these models is beyond the scope of this

thesis. The interested reader is referred to the standard literature [45].

Since the structure of dolphin communication, or lack thereof, is unknown, I use grammar induction to automatically generate hypotheses of the potential structure of dolphin communication. Learning grammatical rules can be achieved using Bayesian model merging to learn a probabilistic context-free grammar as proposed by Stolcke and Omohundro [48] or using a greedy algorithm to induce a context-free grammar such as SEQUITUR [35]. Bayesian model merging starts with a grammar in which each sentence in a data set is represented as a rule. Then new rules are introduced that replace multiple existing rules so the grammar is compressed. The algorithm finds these rules by maximizing the probability with a minimum description length prior probability. In comparison, SEQUITUR starts with a complete text as one rule and introduces new rules that replace multiple substrings. I use alignment-based learning [49] to structure sequences and search for structural rules in the form of regular expressions. Alignment-based learning is based on the idea that parts in a sentence with interchangeable functions will be apparent when aligning sentences.

A recent approach called augmented bag-of-words (ABOW) [3] uses statistics from all three models for activity recognition. ABOW describes a sequence of activities as a histogram including the frequency of each activity (bag-of-words), the frequency of subsequences of activities (n-grams) and regular expressions matching an activity sequence. I adopt this approach for dolphin communication since it gives the flexibility to choose the amount of structural information required in the model. Since it is unclear if dolphin communication is structured, the n-grams or grammar models can be excluded.

CHAPTER III

DISCOVERING PATTERNS IN DOLPHIN COMMUNICATION

In this chapter I describe my approach to frequency shift- and time warp-invariant pattern discovery. First I will formally introduce the problem of finding sequential patterns in dolphin communication. Afterwards, I will describe an algorithm capable of transforming dolphin signal recordings into a novel feature space. The novel feature space will enable easy detection of dolphin signals and comparison of dolphin signals under frequency shifts. The following section will describe why dynamic time warping, piecewise aggregate approximation and hidden Markov models are appropriate models for warp-invariant sequence analysis. In the last section I describe my algorithm for pattern discovery using the definitions and insights described in this chapter.

3.1 Sequential Patterns in Dolphin Communication

Biology researchers capture audible dolphin communication in digital field recordings. An audio file is a digital representation of the recorded sound wave. Since several environmental sounds contribute to the recordings, a common analysis technique is a spectrographic display or spectrogram. A spectrogram for an audio recording can be regarded as a multivariate continuous time series:

$$S = \{s_1, ...s_T\}, s_t \in \mathbb{R}^F \tag{1}$$

Each point in the spectrogram s_{tf} represents the magnitude of frequency f at time t of the original audio wave. Since dolphin signals are inspected traditionally in a spectrogram, and modern speech recognition systems use this representation

as a starting point for further feature extraction, I also search for patterns in the spectrogram. I define a signal pattern as a set of subsequences from several spectrograms that appear similar to each other given a distance function. I define a dolphin communication sequence as a discrete string of signal patterns:

$$P = \{p_1, ...p_T\}, p_i \in \mathbb{P} \tag{2}$$

Each pattern is an element of a global pattern codebook $\mathbb{P}$ shared across all sequences.

In the following I describe an algorithm that can convert a set of spectrograms into a set of dolphin communication sequences using feature learning and pattern discovery.

3.2 Learning Frequency-Invariant Features

In order to enable frequency-invariant comparison of dolphin signals, I learn a set of k feature extractors spanning a k-dimensional feature space. Two dolphin signals that are similar in shape but in different frequency bands should appear close in the novel feature space under Euclidean distance. Furthermore, the feature space should make a distinction between dolphin signals and other underwater noise sources easy. Using these feature extractors I can convert a spectrogram $S = \{s_1, ...s_T\}$ with F dimensions and length T into a time series in the novel feature space $S' = \{s'_1, ...s'_T\}$ with k dimensions and length T.

The algorithm for feature learning clusters small, local regions from the spectrogram using k-means [9] and transforms a novel spectrogram into the feature space using a soft k-means assignment. The soft k-means assignment computes a distance of cluster regions from the spectrogram and then converts these distances into an influence score for each cluster. The final feature space is constructed by max pooling.

I learn the feature extractors from a dataset of dolphin signals with the following

properties:

1. All audio files in the catalog can be categorized into dolphin whistles, burst pulses or noise.

2. All audio files in the catalog include only samples from the categorization.

These properties will help to learn feature extractors that respond solely to dolphin communication. Each example is stored as a short audio snippet. To construct such a dataset, the biologists cut out these examples manually in a way that we can assume that each file only includes dolphin signal. I transform each audio example in the catalog into its spectrogram representation. The main idea is to represent a feature extractor as a square region learned from a spectrogram containing a dolphin signal. Such a region is a local estimate of the spectrogram around its center. For example, a patch centered at a point on a dolphin whistle might capture a small part of an up sweep in frequency. A patch centered around a different location might capture a down sweep. Such a patch can be regarded as a local estimate in the spectrogram. In my experiments a patch represents approximately half a millisecond in time and one kHz in frequency.

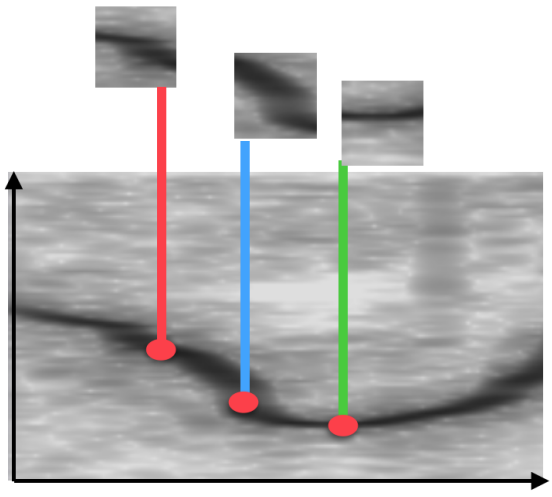


Figure 3: Three patches extracted around a dolphin signal. Two are around different down sweeps and one around a plateau.

Given the spectrograms extracted from my catalog, I extract all patches that fall around dolphin communication. There will be multiple regions that contain up sweeps and down sweeps as well as several regions containing multiple lines as found in burst pulses. I use unsupervised feature learning [9] to form a codebook of regions. I z-normalize each region before proceeding [9]. This process means that from all values in the region, I subtract their mean and divide by their standard deviation.

I then build a codebook of these patches using k-means clustering. The centers of 30 clusters learned from dolphin signal patches are shown in Figure 4.

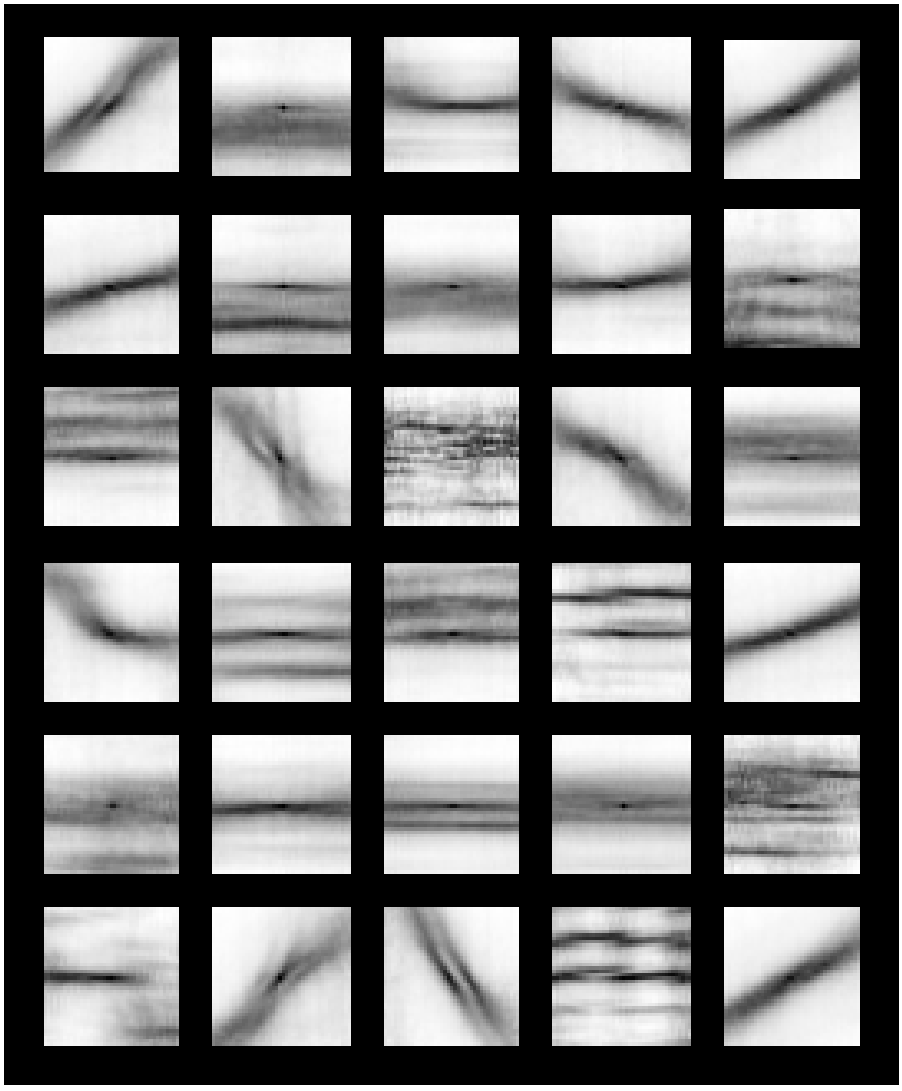


Figure 4: A set of 30 feature extractors learned using k-means.

The resulting codebook represents my feature extractors. A cluster is a square region c with length $2d$. In order to transform a spectrogram into the new feature space spanned by the codebook, I perform the following steps. I place one of the clusters at a point s_{tf} in the spectrogram and compute the distance of the region to the spectrogram area it covers. If I shift the region over the spectrogram and replace each spectrogram point with the distance, I get a new time series $S^c = \{s_1^c ... s_t^c\}$. Each point s_{tf}^c in the new sequence represents the distance of the region c to the spectrogram around the point s_{tf}:

$$s_{tf}^c = \sqrt{\sum_{i=0}^{2d} \sum_{j=0}^{2d} (s_{t-i,f-j} - c_{i+d,j+d})^2} \tag{3}$$

I convert the spectrogram S into the new space S^c for each of the k clusters in the codebook. The result is a set of k new sequences $\{S^{c1} ... S^{ck}\}$. Each entry in the new sequence S_{tf}^{ci} represents the distance of the spectrogram area centered at time t and frequency f to the cluster center c_i. Next I transform the distance representation in a representation capturing the response or influence of each cluster. First I compute the mean of all k distances at every point in the spectrogram:

$$\mu_{tf} = \frac{1}{k} \sum_{i=1}^{k} s_{tf}^{ci} \tag{4}$$

The influence of a cluster at a point in time t and frequency is

$$s_{tf}^c = max(0, \mu_{tf} - s_{tf}^c) \tag{5}$$

If the distance of a cluster is larger than the mean distance at that point, the point is set to zero, so there is no influence. All other cluster influences are proportional to their distances to the cluster. Now each point s_{tf}^c represents the local influence of cluster c to the spectrogram at a point in time t and frequency f. Such an assignment is also called a soft k-means assignment [9]. Finally, we can transform the influence

scores into the new feature space by max pooling. The final feature space is of the same duration as the original spectrogram. The dimension changes to the number of clusters. In order to compute how the influence of each cluster changes over time, I search for the maximum value of each of the k influence sequences at every point in time:

$$s'_{t1} = max_{f=1}^{F} s_{tf}^{c1} \quad (6)$$

$$s'_{t2} = max_{f=1}^{F} s_{tf}^{c2} \quad (7)$$

$$... \quad (8)$$

$$s'_{tk} = max_{f=1}^{F} s_{tf}^{ck} \quad (9)$$

The complete process is shown in Figure 5. As one can see on the top, I visualized the k influence transformations for a whistle. Each point in time and frequency shows the response of a cluster to the underlying whistle. The bottom graphic shows the max pooling process. At every time step the maximum response across all frequencies for each cluster influence is taken as the value in the novel feature space. The result is a k-dimensional time series. Each dimension represents how each cluster's influence changes over time.

The new feature space is frequency-invariant. For example, a cluster center representing a down sweep is shifted over the spectrogram, and the influence is computed at every point. If we pool the responses at every point in time, the frequency at which the maximum response occurred is not represented in the novel feature space. The only information coded in this space is that there was a down sweep at time t with influence s'_{ti}.

I explained how to learn a feature space from a data catalog of local regions extracted from categorized examples. However, I omitted one detail. Earlier I noted that the regions are centered around dolphin signal only, but I did not explain how

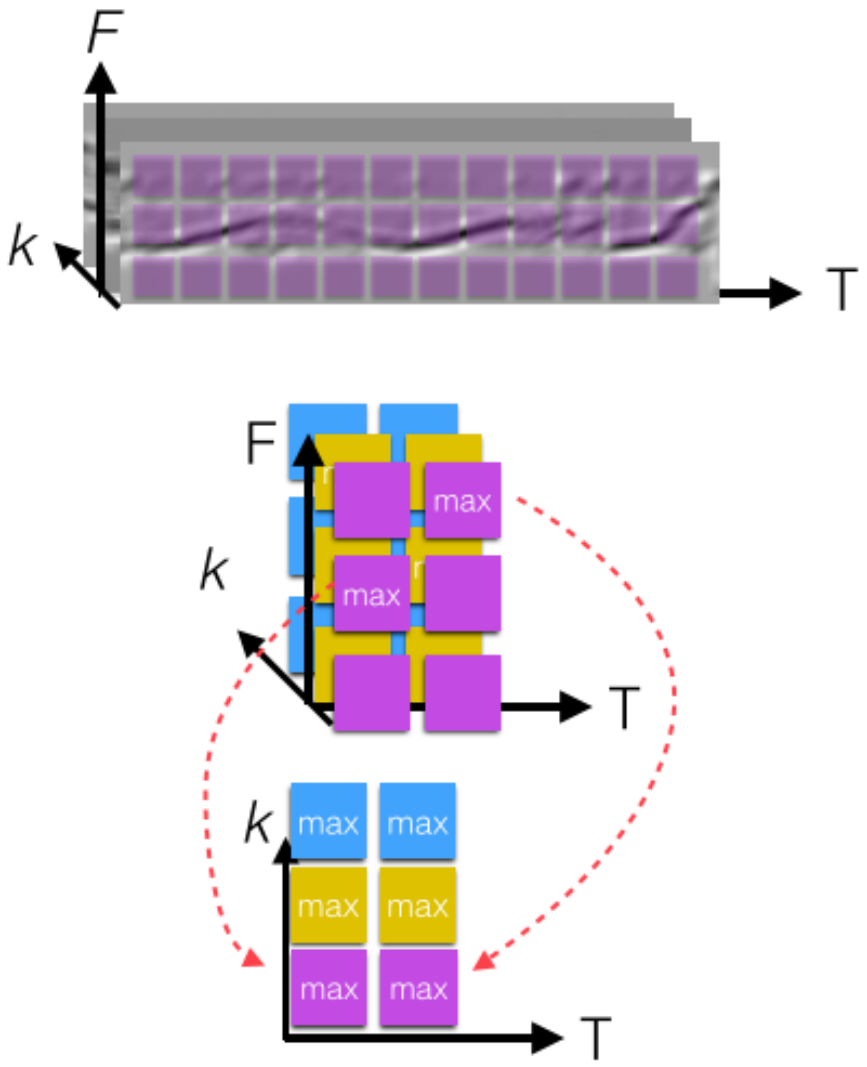


Figure 5: Mapping sequence into feature space.

I ensure they do. In the following I will explain how to center local regions so the region contains dolphin communication rather than other underwater noise. The centering process requires the categorization into whistles and burst pulses. Audio files containing whistles are processed differently than audio files containing echolocation.

3.2.1 Frequency-Invariant Features For Whistles

My goal is to extract regions that contain local spectrogram measurements of dolphin whistles. The categorized catalog contains audio snippets that contain only dolphin communication. The biologists create this catalog by cutting whistles from longer

recordings. In that way, the audio file starts at the start of a dolphin whistle and stops at the end of a dolphin whistle. However, there will still be other noise sources in the spectrogram that happen simultaneously with the whistle. In order to avoid extracting regions that contain local noise estimates, I only extract regions that are centered around the whistle.

I propose to trace the whistle first and then extract regions along the whistle. A dolphin whistle can be thought of as a single oscillator changing frequency over time. In the spectrogram, a whistle will show as a single connected contour. Our regions capturing dolphin communication will be extracted around that contour. A whistle can be extracted by tracing its contour in the spectrogram. In other words, I use the physical properties of a whistle to build a probabilistic model that allows the extraction of clean patches around dolphin whistles.

In noisy environments, tracing is performed using pitch tracking. In the following, I explain a probabilistic pitch tracker for dolphin whistles [22] for the convenience of the reader.

I represent the unknown contour as a path through the spectrogram. At each time step, the path gives the frequency for the trace: $F = f_1 ... f_T$. The first intuition is that the probability of belonging to the contour of a spectrogram entry s_{tf} is dependent on its magnitude. I write this probability as $P(s_{tf})$. I compute the measurement probabilities by normalizing each sample from the spectrogram:

$$P(s_{tf}) = \frac{s_{tf}}{\sum_{i=0}^{D} s_{ti}} \tag{10}$$

This probability model assigns higher probabilities to frequencies with higher magnitude in the spectrogram. I smooth the transition from the last frequency on the trace s_{tf} to the next frequency on the trace s_{t+1f} using a linear predictor.

$$f_{tpredict} = v_{\hat{f}}(t-1) f_{t-1} \tag{11}$$

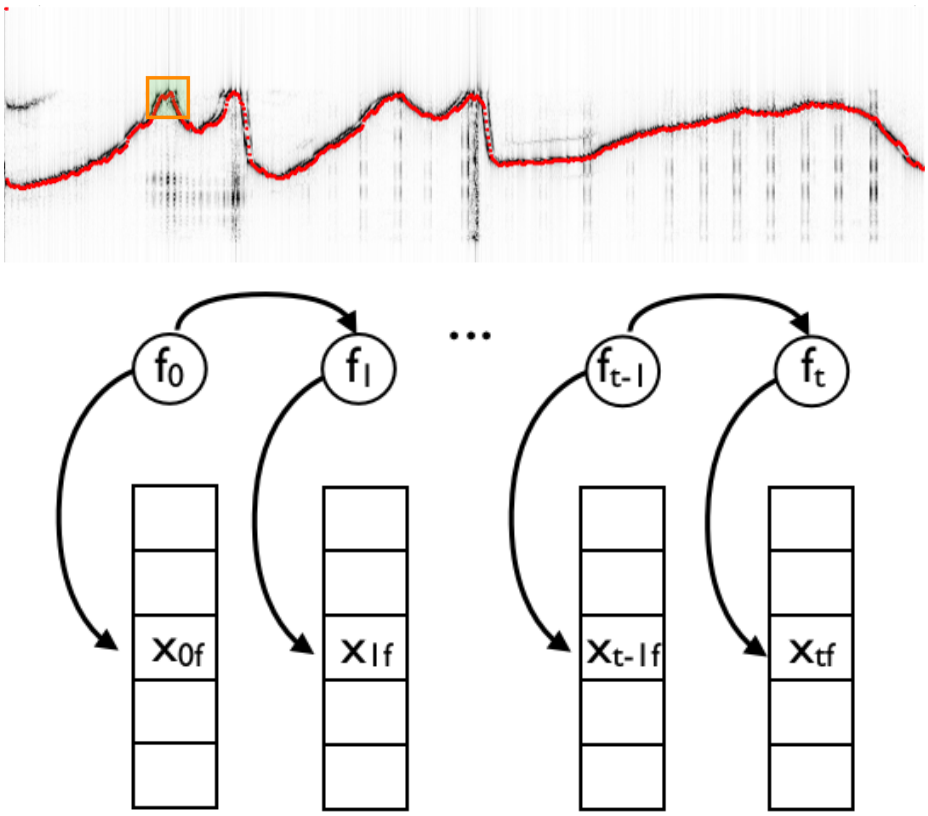


Figure 6: Top: A dolphin whistle showing the extracted contour highlighted in red and a patch around the contour is shown in orange. Bottom: The tracer represented as a probabilistic graphical model.

In order to account for a noisy estimate of the predicted frequency, I model it using a linear Gaussian model centered at the predicted frequency. In other words, I assume Gaussian noise with some chosen variance σ^2 on the current position estimate in the spectrogram.

$$N(f_t|v_{t-1}f_{t-1}, \sigma^2) \tag{12}$$

I can compute this trace recursively using our model. The solution can be found using an algorithm similar to the Viterbi algorithm for hidden Markov model decoding [38]. I solve the following dynamic programming problem to find the frequency trace as the likelihood of frequency f at time t belonging to the trace as defined by

$$\delta_f(t) = P(x_t|f_t) \max_{\hat{f}=1}^{N} \delta_{\hat{f}}(t-1) N(f_t|v_{t-1}\hat{f}_{t-1}, \sigma^2) \tag{13}$$

Furthermore, I store the maximum frequency up to time t in a variable for backtracking purposes. I also compute the velocity estimate from the backtracking variable.

$$ptr_f(t) = argmax_{\hat{f}=1}^{N} \delta_{\hat{f}}(t-1) N(f_t|v_{t-1}\hat{f}_{t-1}, \sigma^2) \tag{14}$$

$$v_f(t) = f - ptr_f(t) \tag{15}$$

Now I can compute a maximum a posteriori trace of the whistle. Then I backtrack from the most likely frequency at time T and follow the backtracking pointer backwards in order to extract the trace. An example of a traced whistle is shown in Figure 6.

3.2.2 Frequency-Invariant Features For Burst Pulses and Echolocation

Signal types besides whistles include multiple lines on top of each other, so it is harder to build a model for them. For example, burst pulse communication shows in the spectrogram as multiple parallel lines with equal spacing between them. The spacing between these lines codes the repetition rate of the clicks. In order to extract clean regions for learning, as in the whistle example, I extract patches around local interest points in the spectrogram.

An interest point in the spectrogram is a point in time and frequency of high magnitude. Formally, an interest point l_{tf} is detected if the spectrogram at time t and frequency f with magnitude s_{tf} is the maximum point in a small neighborhood and greater than a predefined threshold [51]. I also include a local noise estimate in the interest point detection [12]. Assuming stationary noise, the noise $\eta_t f$ is estimated locally as the mean in a plus shaped region with length $2r$ around a potential point:

$$\eta_{tf} = \frac{1}{2r} min\{ \sum_{i=t-r}^{t+r} s_{if}, \sum_{j=f-r}^{f+r} s_{jf} \} \tag{16}$$

An interest point is detected if it is maximal in the plus shaped region or greater than the noise estimate. I extract the patches around these interest points which tend to fall on the harmonic components of a dolphin signal. An example of interest points on multiple burst pulse signals is shown in Figure 7.

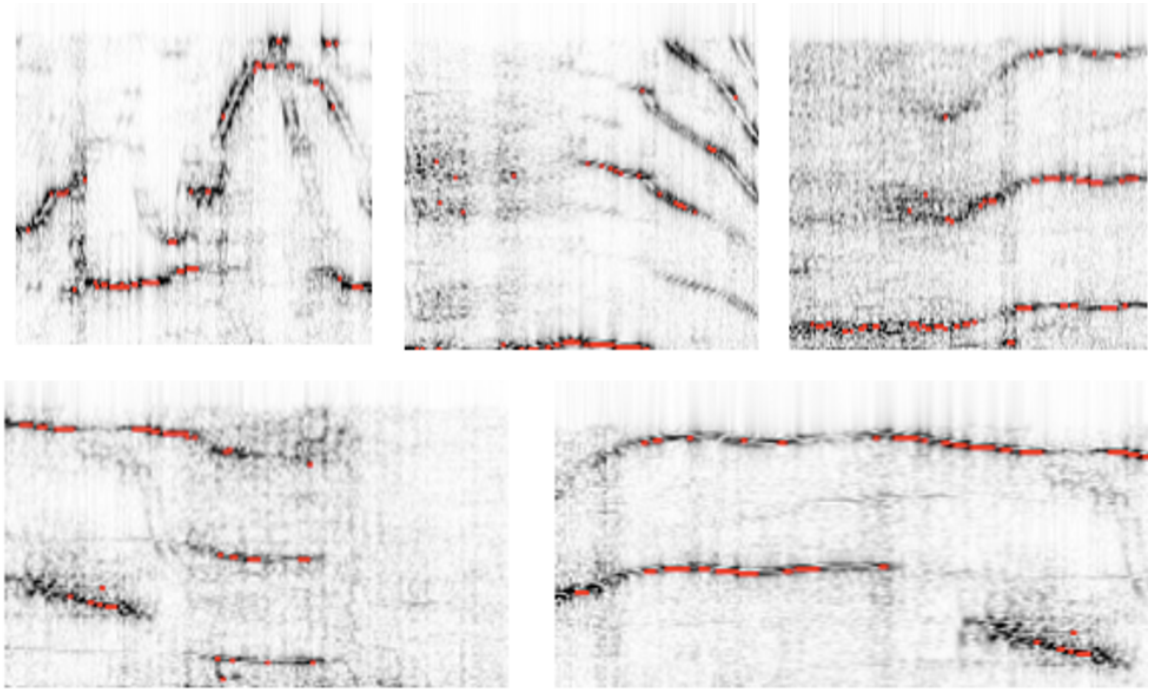


Figure 7: Interest points on a burst pulse signal. The interest points are marked in red.

3.3 Mining Warp-Invariant Patterns

In this section I will describe a general approach to time series analysis with respect to time warping. The goal is to find models and distance functions for time series that account for stretching and shrinking effects.

The first approach introduces the general idea of achieving time warping invariance by dynamic programming. I will explain the concept using the dynamic time warping distance (DTW) [44]. The second section introduces a probabilistic model for time

series called a hidden Markov model (HMM) [38]. I will also point to differences and similarities between hidden Markov models and the dynamic time warping distance. The last section introduces an approximative approach to warp-invariant time series comparisons. I will describe the piecewise aggregate approximation and how it can be used as the basis for further discretization.

3.3.1 Dynamic Time Warping

First I will describe the benefits of comparing dolphin signals using the dynamic time warping distance. Audible dolphin signals with the same shape can undergo time warping transformations when the signals or parts of them are produced at varying speeds. When comparing dolphin signals it is essential to compare signals with respect to deformations based on stretching and shrinking. When computing the distance between two dolphin signals similar in shape but transformed by time warping effects, the distance should still be low.

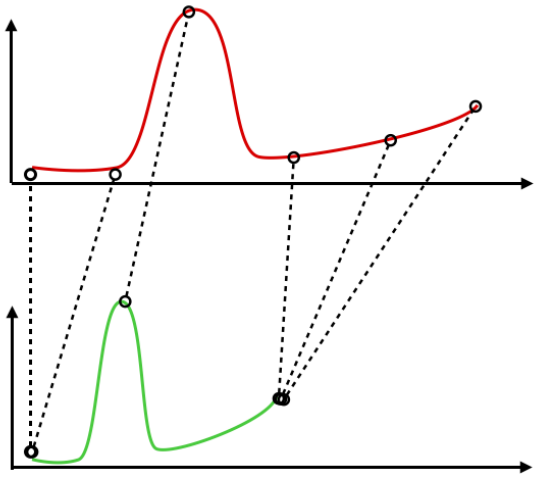


Figure 8: A Dynamic time warping example for two hypothetical time series X (Top) and Y (Bottom). The dashed lines indicate which sample from X aligns to which sample in Y.

For example, if we compute the distance between the two signals $X = \{1, 10, 30, 3, 5, 10\}$ and a shrunk sequence $Y = \{1, 30, 10\}$, a naive comparison approach might be to extend the second sequence to $Y = \{1, 1, 30, 30, 10, 10\}$ and then compute the Euclidean

distance between the two. The resulting distance value is 27. However, the two sequences are similar in shape so we might desire a distance function that takes the warping effect into account. One solution is to compute an alignment between the two sequences first. An alignment assigns each sample in X to a close sample in Y (see Figure 8). If we compute the alignment first and then take the summed distance from sample to sample, we get a distance of 12 for our example. The dynamic time warping distance uses dynamic programming to find an alignment between sequence X and Y such that the summed distance along the alignment is minimal using the following recursion:

$$dtw(X,Y)_{0,0} = 0.0 \tag{17}$$

$$dtw(X,Y)_{i,0} = \infty \tag{18}$$

$$dtw(X,Y)_{0,j} = \infty \tag{19}$$

$$dtw(X,Y)_{i,j} = |X_i - Y_j|_2 + min \begin{cases} dtw(X,Y)_{i-1,j} \\ dtw(X,Y)_{i-1,j-1} \\ dtw(X,Y)_{i,j-1} \end{cases} \tag{20}$$

Dynamic time warping is successfully applied to gesture recognition, activity recognition and speech recognition. Furthermore, the dynamic time warping distance is often used for time series clustering and pattern discovery. In my system, I use the dynamic time warping distance to compare dolphin signals represented in my novel feature space in order to account for time warping effects.

3.3.2 Hidden Markov Models

A hidden Markov model (HMM) has been a probabilistic process that is successfully applied to various domains such as speech and gesture recognition in the past. In the following, I will give a general description of hidden Markov models followed by an explanation of how these models can be used to model a set of dolphin signals and

then how these models account for time warping effects.

I use the standard terminology of an observation and an observation sequence. An observation sequence is a time series, and an observation is one sample at a specific point in time from the time series. For example, our feature space $S' = \{s'_1 ... s'_T\}$ is an observation sequence, and each sample s'_t is an observation. A hidden Markov model is an unobserved first order Markov chain with k states. In the following, I write the Markov chain's current state as y_t. The Markov chain is defined by a transition probability between states $P(y_t = i|y_{t-1} = j)$, also written as a_{ji}. Furthermore, a hidden Markov model is defined by a second function modeling the probability of a sample s'_t belonging to a specific state y_t. This probability function is called the observation distribution: $P(s'_t|y_t = i)$.

Consider the hidden Markov model in Figure 9 as a running example throughout this section. At the top I show the transition function of my hidden Markov model as a sparsely connected graph. Each node in the graph represents a state in the hidden Markov model. Each directed edge in the graph represents a non-zero transition probability between states. Often pattern recognition experts model the temporal dynamics of a hidden Markov model by defining the connectivity between states. In other words, the experts decide which transitions can have a non-zero probability. Below the transition function I visualize the observation function. For continuous multivariate observations, a common observation distribution is a multivariate normal distribution. I visualize the mean vectors of the observation distribution, one for each state. In my novel feature space, each dimension represents the influence of a feature extractor. The influence is indicated by the height of the bar. In this example, the first state represents a down sweep and a plateau of the whistle, the second state defines a steep up sweep, and the third state represents a down sweep as modeled by the observation function. The transition function automatically codes an order. In our example, we have to transition through the first state before we can transition to

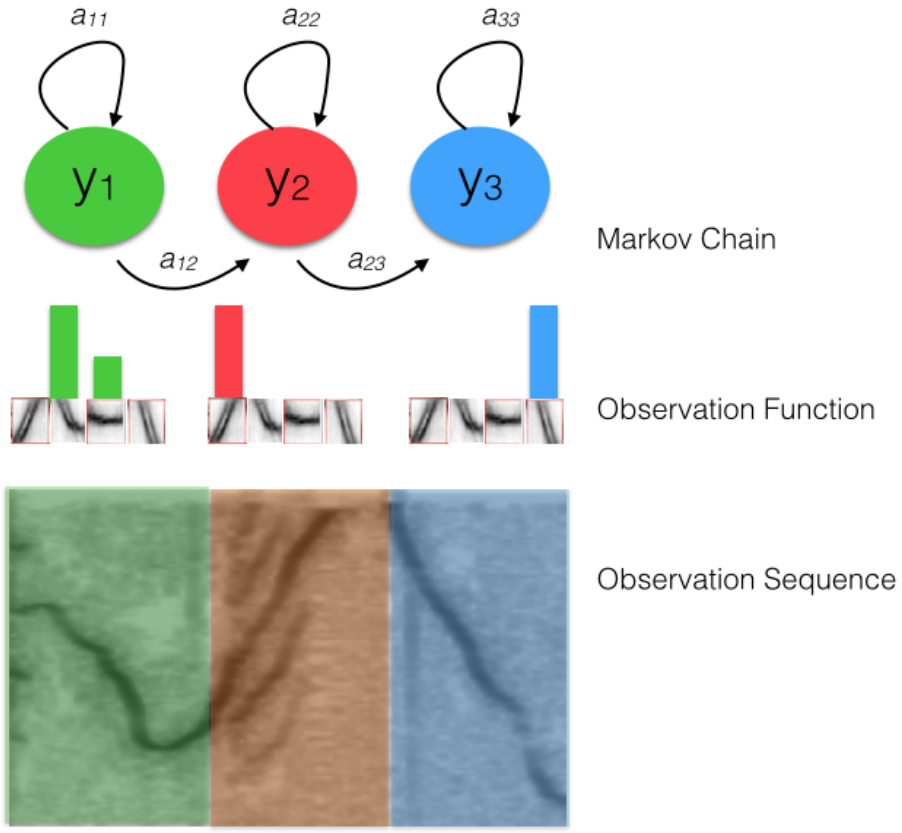


Figure 9: A hidden Markov model with three states. Top: The Markov chain defining the transition function. Middle: The observation function for a feature space with four dimensions. Each of the histograms represents a mean vector of a multivariate Gaussian. Each dimension in the mean vector represents the influence of a cluster. Bottom: A hypothetical alignment of a whistle to the model's states. The visualization shows the spectrogram of the whistle. The colors represent the assignment of each sample to a state.

the second state, and we have to transition to the second state before we can transition to the third. Together with the observation function, the HMM models the sequence, "down/slow, up/steep, down/steep." The expected duration we stay in each state is coded in the self transitions. The expected duration spent in a state is equal to the expected value of the geometric distribution: $\frac{1}{1-a_{ji}}$. At the bottom of Figure 9, we see the spectrogram of a whistle. Converting the spectrogram into the novel feature space enables an alignment of the observation sequence into the HMM's state space. Such an alignment can be constructed using the Viterbi algorithm. While dynamic

time warping aligns all samples of a time series to a sample of another time series, the Viterbi algorithm aligns each sample to a state in the HMM using the following dynamic programming recursion:

$$v(0,i) = \pi_i * N(s'_0|\mu_i, \sigma_i^2) \quad (21)$$

$$v(t,i) = N(s'_t|\mu_i, \sigma_i^2) * max_j a_{ji} v(t-1,j) \quad (22)$$

The normal distribution representing the observation function is written as $N(s'_t|\mu_i, \sigma_i^2)$. Furthermore, π_i represents the probability of starting the alignment in state i. While the dynamic time warping distance finds the alignment that minimizes the Euclidean distance between the samples of two sequences, the Viterbi algorithm creates the alignment from a sequence to states of the hidden Markov model that maximizes the probability along the alignment path. The transition probability and the observation probability are both taken into account when constructing the path. Deviations from the expected path given by the transition probabilities happen when the observation probability for another path gets higher.

The parameters of a hidden Markov model can be estimated from a set of observation sequences using an algorithm called Baum-Welch. Describing Baum-Welch is beyond the scope of this description, and the interested reader is referred to Rabiner's work [38].

One advantage of using hidden Markov models is that they can be combined into larger ones (see Figure 10). For example, if we have two hidden Markov models each modeling a different dolphin signal pattern, we can construct a new hidden Markov model combining them. By connecting each end state of the two models to each start state, we can model observation sequences containing both patterns in sequence. For example, a common solution in speech recognition is to construct a hidden Markov model for each word. In order to recognize whole sentences, all the word models are

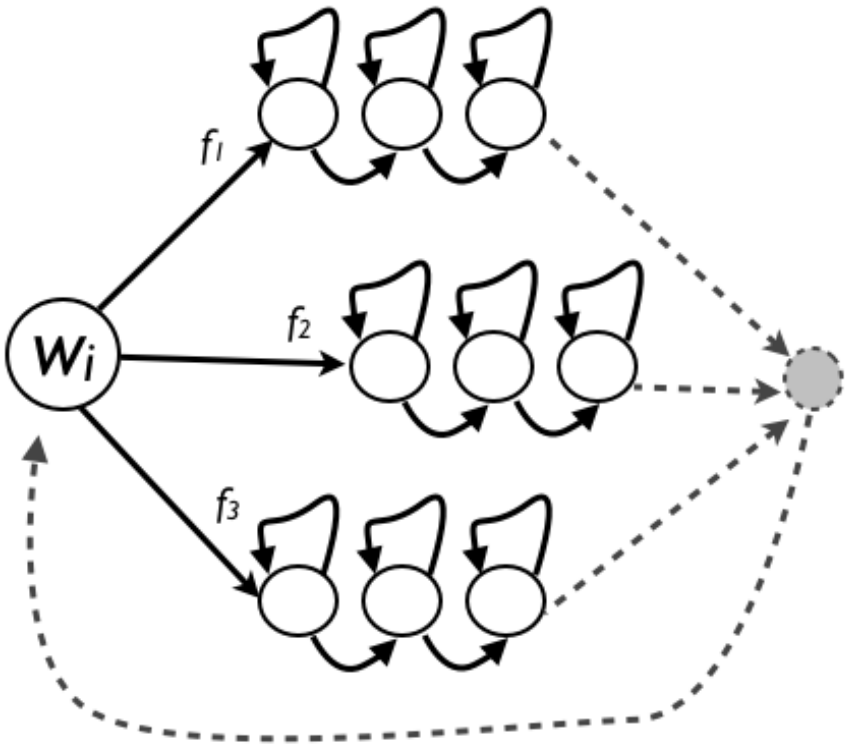


Figure 10: Multiple hidden Markov models combined into a joined model by connecting the end states to each start state.

combined. The parameters for each word model are trained from multiple audio files for each word.

Later in this chapter, I will use these combined models called mixtures of hidden Markov models to jointly explain all dolphin signal patterns.

3.3.3 Piecewise Aggregate Approximation

The last approach to warp-invariant pattern comparison is based on an approximation. While the hidden Markov models and the dynamic time warping distance approach the time warping effects using alignments, this approach is based on compressing all sequences to the same length, discretizing the compressed representation and using string equality for comparison.

I compress a sequence using the piecewise aggregate approximation (PAA) [19].

The PAA compresses a time series by splitting the sequence into k equally sized bins. The bin size to compress a sequence of length T is $w = \frac{T}{k}$. The compressed sequence $C = \{c_1...c_k\}$ for a dolphin signal in the novel feature space $S' = \{s'_1...s'_T\}$ is calculated as

$$c_i = \frac{1}{w} \sum_{t=i*w}^{(i+1)*w} s'_t \tag{23}$$

The new feature space represents the influence of a patch in the codebook over time. The compressed representation aggregates the influence in each of the w bins. My discretization uses the index of the codebook entry with the maximum influence in each bin as a symbol, resulting in a string X of length w:

$$X = \{argmax^D_{i=1}(c_{1d})...argmax^D_{i=1}(c_{wd})\} \tag{24}$$

The comparison of sequences in the discrete space is based on string comparisons. Two sequences are considered similar if their discrete representations are equal. The complete representation is shown in Figure 11.

3.4 Warp- and Frequency-Invariant Pattern Discovery in Dolphin Communication

In the last section I will describe the complete algorithm to pattern discovery. Leveraging feature invariance and warp invariance, I will present an algorithm that takes a spectrogram $S = \{s_1...s_T\}$ of dolphin communication as the input and outputs a discrete dolphin communication sequence $P = \{p_1, ...p_T\}$.

My goal is to learn a representation in which all patterns as well as the underwater noise sources are modeled in a probabilistic model. I learn this model in three steps (see Figure 12).

1. Learn the initial segmentation in the novel feature space

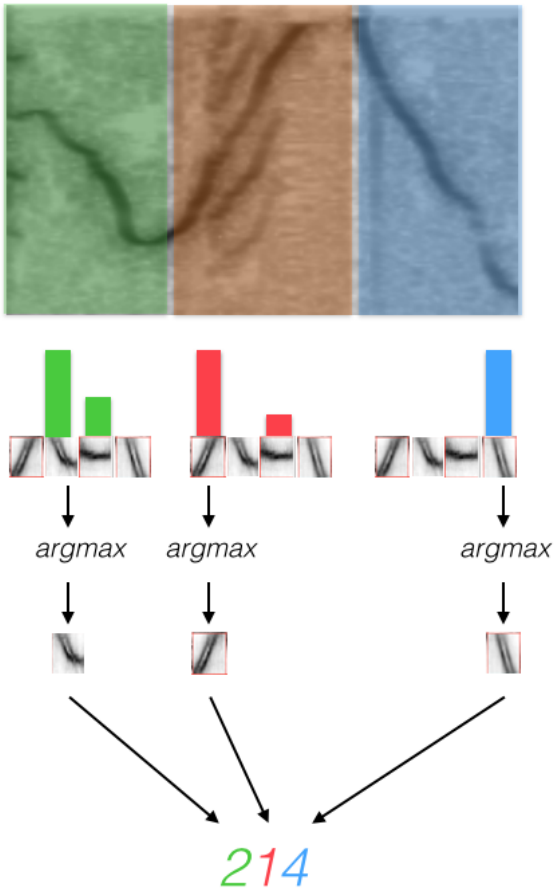


Figure 11: A time series is split into three equal regions. The compressed representation is shown in the novel feature space. I calculate a discrete representation from the compressed representation using maximum influence.

2. Cluster dolphin signals into patterns

3. Learn a probabilistic joint model of all patterns and noise

In the first step, I convert the spectrogram $S = \{s_1...s_T\}$ into the novel feature space $S' = \{s'_1..s'_T\}$. The novel feature space represents the influence of local dolphin signal movement patterns. Furthermore, I classify each sample in the sequences as dolphin signal or noise using a random forest [5]. Using the classification results I extract regions of consecutive samples classified as dolphin signal. I extract sliding windows from these regions and cluster the windows into patterns. After the clustering is done, I learn a hidden Markov model for each cluster. Furthermore, I learn a

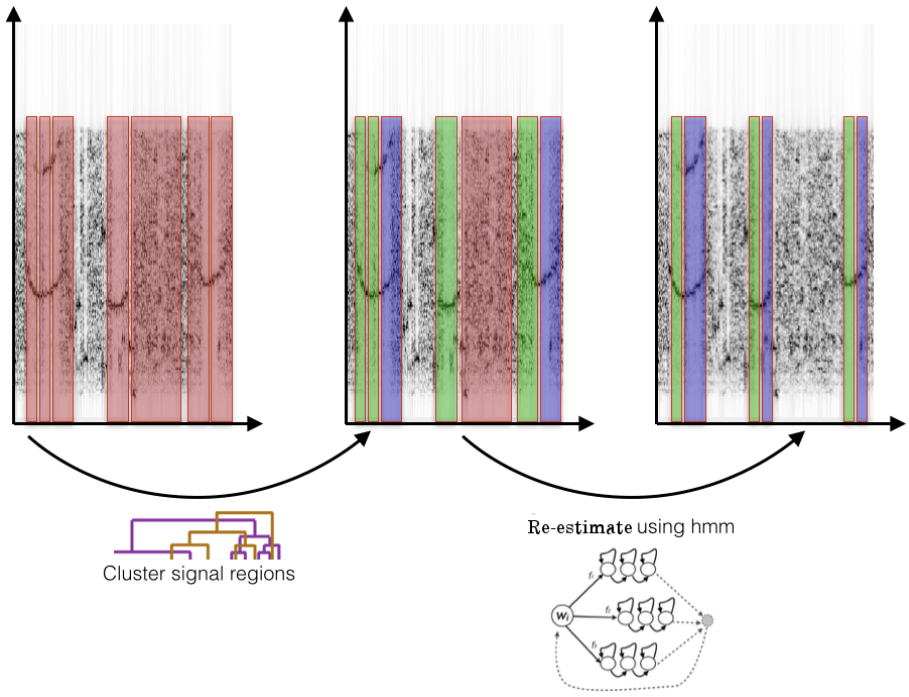


Figure 12: Overview of the data mining system. After identifying signal and noise regions using a binary classifier, the resulting regions are clustered. For the final estimate I train a left-to-right hidden Markov model for each cluster. Together the models form a mixture of hidden Markov models. Decoding each sequence with that mixture gives a smooth segmentation, fixing boundary errors and noise assignments.

mixture of Gaussian from the samples classified as noise. I then combine the resulting models and the mixture into a joint hidden Markov model. In the following, I will give the details for two clustering algorithms and how to combine their results into the final hidden Markov model.

The resulting hidden Markov model can be regarded as the pattern codebook. The model contains every pattern that can occur in each spectrogram. A final shared pattern codebook is responsible for the conversion of a piece of dolphin communication in the feature space $S' = \{s'_1..s'_T\}$ into a dolphin communication sequence $P = \{p_1..p_N\}$.

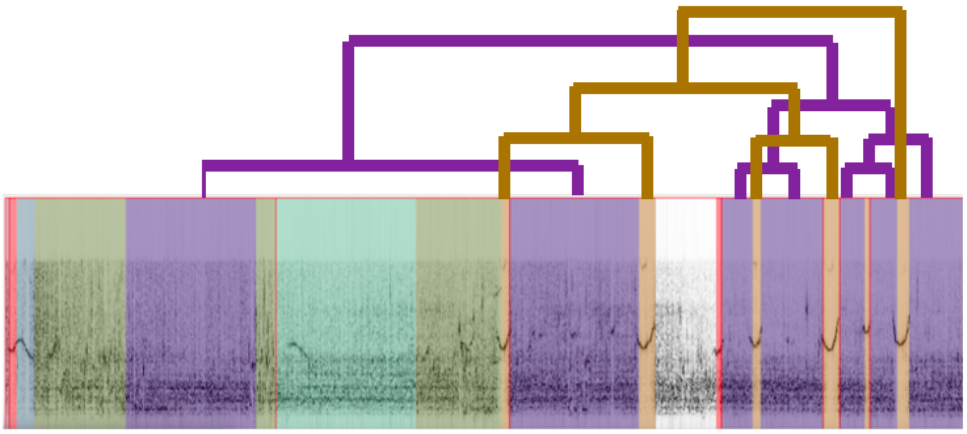


Figure 13: Hierarchical clustering of regions in dolphin communication.

3.4.1 An Exact Pattern Model

The first clustering algorithm clusters the sliding windows using agglomerative clustering under the dynamic time warping distance (see Figure 13). In agglomerative clustering, all windows initially represent their own cluster. In each clustering step, the algorithm merges the two closest clusters. I use the average linkage as the closeness between clusters. Since I am using dynamic time warping, the average linkage between two clusters c_1 and c_2 is the average dynamic time warping distance between windows in cluster c_1 and cluster c_2. I stop merging at a user-defined maximum number of clusters. I proceed by learning one left-to-right hidden Markov model from each cluster using the Baum-Welch algorithm. Since the maximum number of patterns will lead to over-segmentation, I apply greedy mixture learning [34].

Greedy mixture learning starts with a one-state hidden Markov model representing the noise. The observation distribution is the mixture of Gaussians estimated from the noise samples. We then greedily add the pattern model to the mixture that maximizes the likelihood for all data. If the increase in likelihood is not sufficiently large, the algorithm returns the mixture. Now I can decode all communication sequences in my sequence database using the Viterbi algorithm. By assigning each sample to the pattern indicated by the Viterbi path, I achieve a segmentation into patterns:

Algorithm 1: Segmented Clustering algorithm.

Data: A set of clusters C from a set of dolphin communication sequences S. A threshold to define when to stop.
Result: A mixture of hidden Markov models M
A mixture containing only the silence model M;
$lastLL = -\infty$;
while $|C| > 0$ **do**
  $maxCluster = NULL$;
  $maxModel = NULL$;
  $maxLL = \infty$;
  for *each cluster* c *in* C **do**
    train a hidden Markov model m from data in c;
    compute log likelihood $LL(S|\{m\} \cup M)$ all models using the mixture and the new model;
    if $LL(S|\{m\} \cup M) > maxLL$ **then**
      $maxModel = m$;
      $maxCluster = c$;
      $maxLL = LL$;
    end
  end
  $M = M \cup maxModel$;
  $C = C \setminus \{maxCluster\}$;
  if $maxLL < lastLL$ **then**
    break;
  end
  $lastLL = maxLL$;
end
return *mixture* M;

$P = \{p_1, ...p_T\}, p_i \in \{1...N\}$.

If my feature codebook has D components, the first sequence is N samples long and the second sequence is M samples long, dynamic time warping has to compute the Euclidean distance in D dimensions at every step, resulting in a worst-case complexity of $O(N * M * D)$ for each comparison. If my dataset contains T sliding windows, the worst-case complexity to compare all sequences to all other sequences is $O(T^2 * N * M * D)$. Since the algorithm is very slow, I propose a faster, approximate clustering algorithm, based on discretizing the regions first and then hashing similar sequences together.

3.4.2 An Approximate Pattern Model

The approximate model is quite simple. I convert all sliding windows into a discrete representation using the compression approach. I continue to insert all sequences into a hash table. All collisions in a particular entry in the hash table represent a different cluster. I use a user-defined threshold on the cluster size and train a hidden Markov model from all sequences of clusters that are large enough. I proceed by combining all these models into a mixture.

Converting the dolphin sequences into pattern strings is equal to the exact procedure. I decode all communication sequences in my sequence database using the Viterbi algorithm. Each sample is assigned to the pattern indicated by the Viterbi path, and I achieve a segmentation into patterns: $P = \{p_1, ...p_T\}, p_i \in \{1...N\}$.

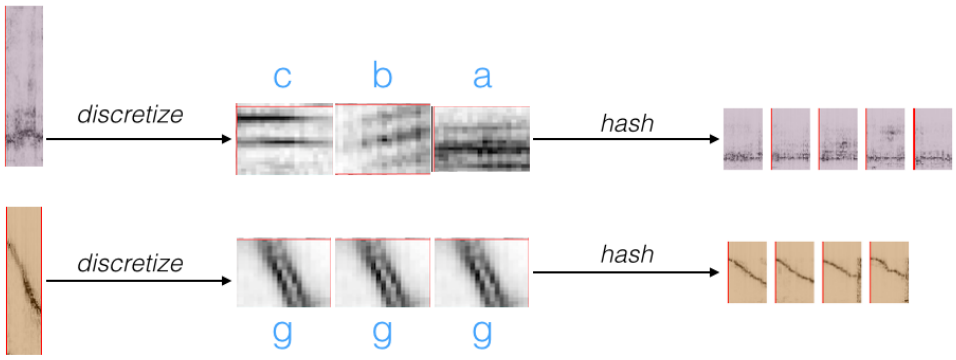


Figure 14: The approximate discovery process: First a signal in the new feature space is compressed and discretized. In the resulting string each symbol represents one of the feature extractors. The discrete strings are inserted into a hash table. All sequences hashed to the same table's entry are considered to belong to the same cluster.

The approximate algorithm has the advantage of finding patterns in a very fast manner. While the algorithms for the exact approach run in polynomial time, the system can be very slow. The exact algorithm finds the clusters in linear time. Furthermore, since the cluster size determines how many models we have in the final mixture, we do not have to use the more expensive greedy mixture learning procedure.

At this point the system is able to convert a spectrogram of dolphin signals into a novel feature space and search for patterns in it. The discovered patterns are invariant to frequency shifts and time warping transformations. Effectively, the algorithms convert a spectrogram into the desired dolphin communication sequence. In the next chapter, I will explain how to compute statistics from a set of dolphin communication sequences.

CHAPTER IV

A DATA MINING SYSTEM FOR DOLPHIN COMMUNICATION ANALYSIS

In the last chapter, I described how to find patterns in a set of spectrograms containing dolphin communication. Formally, I convert each spectrogram $S = \{s_1...s_T\}$ into a string of patterns $P = \{p_1...p_N\}$ called a dolphin communication sequence. In this chapter, I describe my approach to building statistical models from dolphin communication sequences. Furthermore, I show how the resulting statistics can be used to label unseen data with behavior annotations and how to run comparative statistics between different behavioral contexts.

My approach to statistical modeling uses discrete distribution estimated as counts of patterns, n-grams of patterns and sequential rules of patterns in the form of regular expressions.

4.1 Models of Dolphin Communication

My communication model is based on distribution of pattern counts. It is inspired by successful models from information retrieval. In information retrieval, a document is described by word frequencies or the frequencies of features extracted by sentences. In the following, I will describe the statistics I can calculate from a set of dolphin communication sequences. Formally, a dolphin communication sequence is a string of patterns $P = \{p_1...p_N\}$. Each pattern in the string is an element of all possible patterns in the database $\mathbb{P}$ of size k. A dataset is a set of patterns $D = \{P_1...P_M\}$. For example, if the biologists collect a set of audio files containing 10 different pieces of dolphin communication and I run the pattern discovery algorithm, I get a set of

10 pattern strings. The statistical methods I use here have been successfully applied activity recognition in a method called "augmented bag-of-words" [3].

Augmented bag-of-words captures long-term interactions between symbols in a sequence by using a set of regular expressions from the strings in the data set. The short-term interactions are captured using a set of n-grams. For each sequence, I count the number of times each symbol occurs (unigrams), the number of time each n-gram occurs and the number of times each rule matches the sequence (see Figure 15).

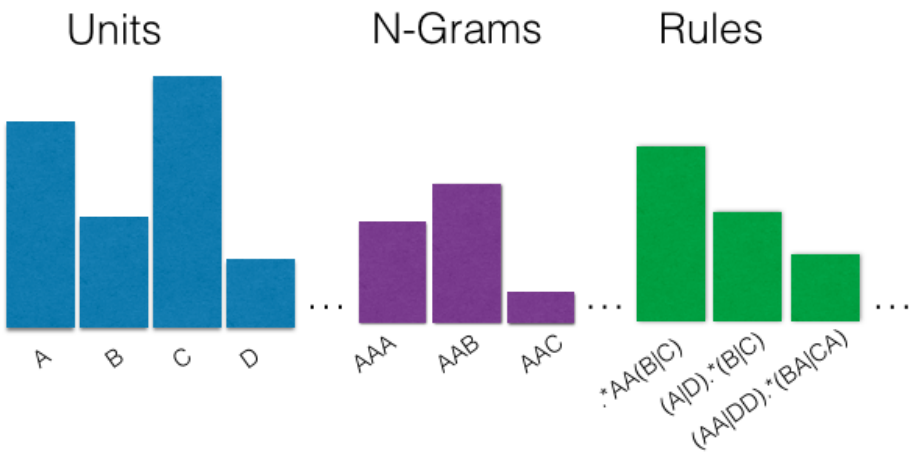


Figure 15: Communication model constructed as a combination of single units, n-grams and rules.

I describe these models in detail below.

4.1.1 Counting Patterns

The simplest approach to building the statistics is to build a histogram of pattern occurrences for every dolphin communication sequence.

$$p(x|P_1) = \frac{count(x, P_1)}{\sum_{y \in D} count(y, P_1)} \tag{25}$$

The probability is simply the normalized count of each pattern in the sequence. The probabilities of each pattern are used as a descriptor for the sequence. In other

words, in this approach I disregard temporal information. The order in which the patterns occur has no effect in this representation. In other words, the global sentence structure is lost in this representation. For example, the histogram of the sequence "aabaabaa" is $\{a : 0.75, b : 0.25\}$. The two following distributions help to overcome this problem.

4.1.2 Pattern N-Grams

The second approach calculates a probability distribution of subsequences with length N called an n-gram. Extracting all sequences of length N with a sliding window and building a histogram of all the windows in a sequence results in a n-gram probability distribution:

$$p(x_t, x_{t-1}...x_{t-n|P_1}) = \frac{count(x_t, x_{t-1}...x_{t-n}, P_1)}{\sum_{y_1...y_N} count(y_1...y_N, P_1)} \tag{26}$$

In this statistic, the probability is defined as the normalized count of an n-gram. An n-gram distribution can be regarded as a Markov chain of $n - th$ order. It represents sequential information in a local context. A communication sequence is now represented as a histogram of n-grams. For example, extracting all the sliding windows of size two from the sequence "aabaabaa" results in the following histogram: $\{aa : 0.42, ab : 0.29, ba : 0.29\}$. While the simple pattern count does not account for temporal structure, the n-grams capture local information.

4.1.3 Pattern Rules

The previous statistics model the frequency of patterns or the frequency of local pattern sequences. The last statistic aims to model the global structure of patterns. I model the global structure of a communication sequence as regular expressions.

A regular expression is a sequence of symbols defining a search pattern. For example the string $ab[a-Z]*(b|cd)$ defines a search pattern in which the string "ab"

is followed by a string of any characters "[a-Z]*" with any length. Then the string ends with either "b" or "cd." Example strings that match this search patterns are

1. abaaaaaab
2. abcccbbbbcd
3. abaababababacdb

All strings start with the required "ab" sequence. In the first string, the middle sequence "aaaaaa" is mapped to the string of any character "[a-Z]*" and the string ends with a "b." In the second sequence, the any character sequence is "cccbbbb" and the string end with "bc." In the last string, the middle part is "aababababacd" and it ends with a "b." My goal is to extract regular expressions automatically from a database of dolphin communication patterns. For every dolphin communication sequence, I find all regular expressions that match the sequence. The matching expressions are used as my representation for the sequence.

In the following, I will describe how to learn a set of regular expressions from a database of dolphin communication patterns using an algorithm called "alignment-based learning" [49]. The resulting regular expressions support regions where no character matches and regions with an OR.

Alignment-based learning is used to learn context-free grammars from a text corpus. Alignment-based learning assumes that parts of a sentence with the same function can be replaced by each other. As the name suggests, these replacements are found using an alignment. Instead, I do not aim to learn a context-free grammar from alignments but regular expressions.

In the last chapter, I introduced the idea of using alignments of continuous sequences to account for the time warp. Here I construct alignments between pairs of symbolic sequences.

A pairwise alignment between two sequences $X = x_1...x_i...x_N$ and $Y = y_1....y_j...y_M$ can be achieved by a series of insertion, deletion, substitution and match errors. An insertion error at position x_i means the symbol is not present in y_i. A deletion error means the symbol is present in y_i but not x_i. A substitution error means the symbol at x_i is different from the symbol at y_i. A match is no error, meaning the symbol x_i and y_i are the same.

I use the Needlemann-Wunsh algorithm to construct the alignments [13]. The algorithm is very similar to the dynamic time warping algorithm.

$$nw(i,j) = max \begin{cases} nw(i-1,j) & + & \delta \\ nw(i-1,j-1) & + & s(x_i, y_i) \\ nw(i,j-1) & + & \delta \end{cases} \tag{27}$$

In the above example, δ is a penalty for insertion or deletion errors. Furthermore, the function $s()$ returns a penalty for each pair of characters.

The recursion is initialized as $nw(i,0) = -i * \delta$ and $nw(0,j) = -j * \delta$. Backtracking through the dynamic programming solution allows it to find the alignment. From the alignment, I can retrieve the insertions, deletions and match operations. In bioinformatics, the substitution between symbols is estimated from substitution statistics collected from existing manually created alignments. Since I do not have access to manual alignments, I set the substitution cost to "1" if the two symbols match and to "-1" if the two symbols do not match. Furthermore, I set the insertion and deletion penalty to "-1." In other words, all errors are penalized with "-1" and all matches get a positive score of "1".

I use the above procedure between two sequences to extract regular expressions. I construct a regular expression from a pairwise alignment and use regions of matches and substitutions as evidence for parts of the signal that occur across multiple sequences. I construct the regular expression in the following manner. All regions of

matches are unchanged. I replace all regions of insertions and deletions with a sequence of filler symbols of undefined length. In regular expression notation such a sequence is written as $[a - Z]*$. All substitutions are replaced by an OR operator. For example, if one substitution region is "abc" in one sequence and "def" in the other, we define that the regular expression can match either. The regular expression notation is $(abc|def)$. A sample alignment and its regular expression are shown in Figure 16.

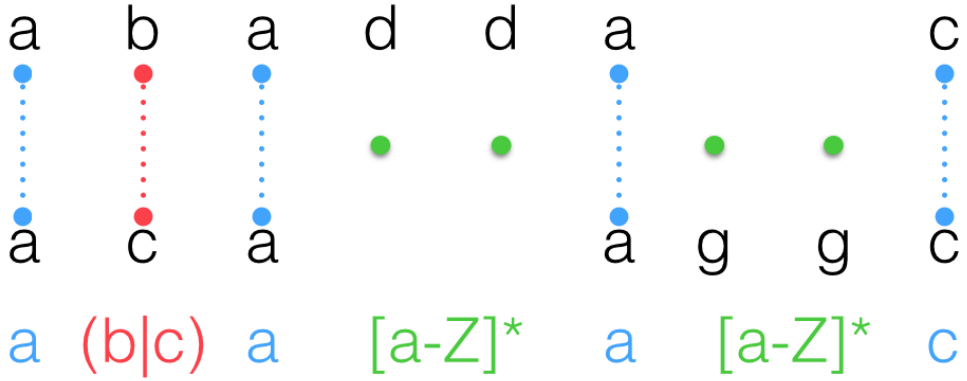


Figure 16: Top: A hypothetical alignment example. Matches are highlighted in blue, substitutions in red, deletions and insertions in green. Bottom: The resulting regular expression.

Now I build a set of all regular expressions from the dataset. Then, I align each sequence to each other sequence and extract the regular expression. I add a regular expression to the set of regular expressions if it matches more sequences than a predefined threshold.

In order to represent a sequence using the new rule set, I calculate if a regular expression matches a sequence in the data set. The result is a binary vector with all fields set to one that represent a matching regular expression.

In Figure 17 I visualize some of the regular expressions extracted for our experiments.

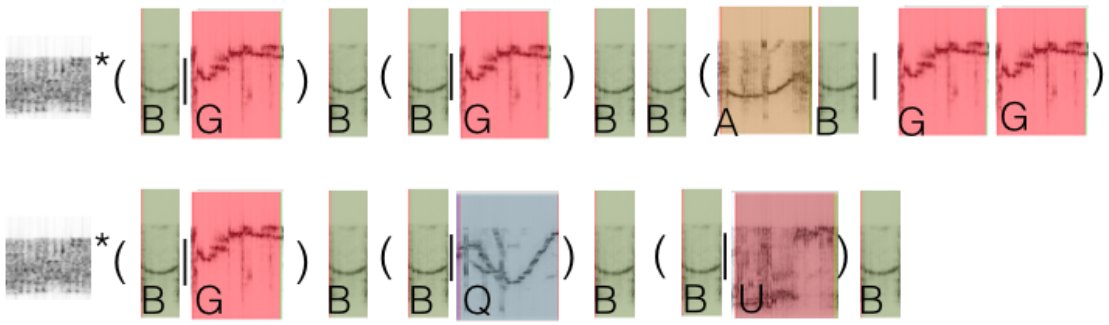


Figure 17: Two rules extracted from our data set. $(x|y)$ represents OR and $*$ represents a repetition of the previous symbol as often as needed to match the rule.

4.2 Annotation of Dolphin Communication

As discussed Section 1.3, dolphin researchers often annotate their video databases with tags describing typical animal behavior, dolphin identifiers and sound categories. The sound category annotations are the same as I use to learn the codebook. The goal is to annotate the audio files with tags from visual behavior and dolphin identifiers (see Table 1 in Chapter I). Researchers create the dolphin identifiers using visual features in the video. Often the spots on a dolphin or scars can identify a dolphin. An example of a typical dolphin behavior might be a dolphin slapping another with its tail. Another example is two dolphins swimming head-to-head. These examples are shown in Figure 18. The annotation performance using the patterns is interesting in itself. Performing annotation experiments with ground truth data can hint towards the quality of the discovered patterns. High performance for observations from the video using patterns found in the audio stream indicates that the patterns have actual meaning. For example, if whistle patterns from the audio stream predict dolphin IDs from the video, the system might have discovered a dolphin's signature whistle.

In the previous section, I showed how to extract several statistics that describe different aspects of dolphin communication. A dolphin communication pattern $P = \{p_1...p_N\}$ can be described as a discrete probability distribution $p(x|P)$. Each entry in the probability distribution for that pattern can be a pattern count, an n-gram

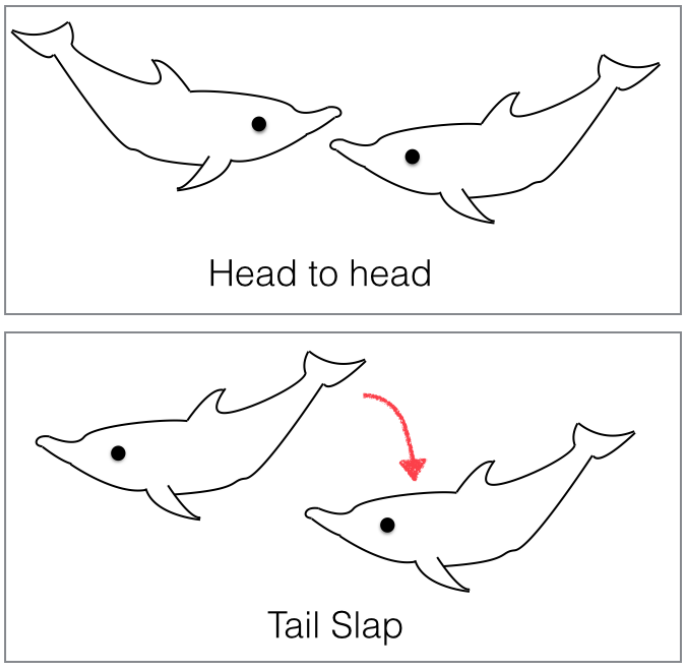


Figure 18: Two examples of annotated behavior. Top: Two dolphins swimming head-to-head. Bottom: A dolphin slapping another with its tail. Reproduced with permission from Miles [31].

count or a regular expression count. In the following, I will describe two methods to annotate dolphin communication sequences automatically using their pattern probabilities. The datasets I use are regions in the audio files with a dense coverage of dolphin signals annotated with such annotations. Formally, I define a label set $Y = \{t_1 ... t_m\}$ associated with a dolphin communication sequence. In the following, I assume I extracted some statistics from these sequences resulting in vectors of statistics $X = \{x_1 ... x_N\}$. Each entry might be the frequency of an n-gram, a pattern or a rule.

For example, a data set might be:

1. $X_1 = \{x_{11} ... x_{1N}\}$, $Y_1 = \{t_1, t_2\}$
2. $X_2 = \{x_{21} ... x_{2N}\}$, $Y_2 = \{t_3\}$
3. $X_3 = \{x_{31} ... x_{3N}\}$, $Y_3 = \{t_4, t_2, t_1\}$

4. $X_4 = \{x_{41}...x_{4N}\}$, $Y_4 = \{t_3, t_1\}$

5. $X_5 = \{x_{51}...x_{5N}\}$, $Y_5 = \{t_1, t_4, t_2\}$

The goal is to convert dense regions into communication sequences and extract statistics. In the following, I will describe how to annotate unlabeled dolphin sequences. The use of these annotations can be to auto-annotate new field recordings and also to evaluate the statistical models. If one statistical model is more accurate than another, the researchers can infer that the one model is a better representation based on their labeling. For example, if a model of n-grams results in 80% annotation performance and another model using combined n-grams and pattern counts yields 90% annotation performance, one inference might be that the second model is better. Furthermore, computing which tags a model can predict more accurately might give an insight into the structure of a specific context. For example, if the rule statistics are very accurate with tags such as "head-to-head" or "tail slap," the researchers might infer that aggressive behavior or play behavior have sequential components. The two tags annotate behavior associated with playful and aggressive social context, and the rules capture long-term sequential patterns. Last but not least, such an annotation system can be used to search for specific annotations in a database of audible dolphin communication. For example, if a biologist wants to find several audio files containing the dolphin "Bishu," such a search system could help. The goal is to achieve high precision with the annotation algorithm in order to reduce the number of documents falsely returned.

4.2.1 K-Nearest Neighbor in Semantic Spaces

The first algorithm I use to annotate dolphin communication sequences is adopted from image annotation [50]. Image annotation and dolphin sequence annotation are similar since there can be multiple tags per sequence or image, so an out-of-the-box classifier can not be applied directly. I annotate each unseen sequence using a

modified version of the k-nearest neighbor algorithm (KNN).

For an unseen dolphin communication sequence, I compute the probability of each tag as described in Verma and Jawahar [50]. In the first step, I group all of our training examples by tags. For m tags $t_1...t_m$, I will have m sets $I_1...I_m$ of examples. Since each example can have multiple tags, these sets will overlap. The resulting sets in our example are:

1. $t_1 : I_1 = \{X_1, X_3, X_4, X_5\}$
2. $t_2 : I_2 = \{X_1, X_3, X_5\}$
3. $t_3 : I_3 = \{X_2, X_4\}$
4. $t_4 : I_4 = \{X_5\}$

For an unannotated sequence Q, I extract the statistics first. Then I compute the k-nearest neighbors in each set I_i, resulting in a set of $k * m$ neighbors. An example is given in Figure 19.

For each of the m tags, I compute the probability of the query showing a tag as exponentially decreasing with the distance to instances showing that tag.

$$P(Q|t_j) \propto \sum_{i=1}^{k*m} e^{-|X_i,Q|_2 * \mathbb{I}(t_j \in Y_i)} \tag{28}$$

Now I annotate a region by computing the above probabilities for each tag and annotate a region with all tags showing a probability greater than a predefined threshold.

$$P(Q|t_j) > \theta \tag{29}$$

This approach generates a probability distribution over all possible tags for each new instance individually. The tags are assigned by applying a user-defined threshold. One problem is that the number of neighbors and the threshold have to be chosen

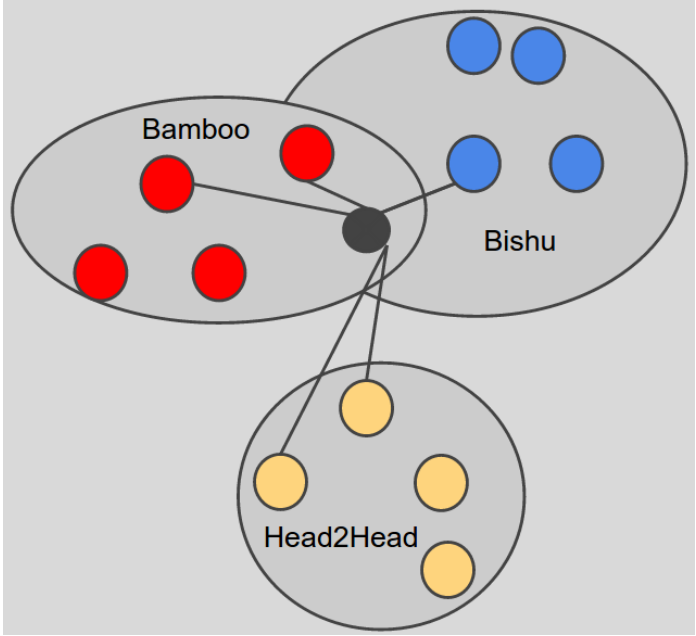


Figure 19: An example evaluation of the semantic two-NN algorithm. The query instance is shown as a black circle. The dataset is segmented into three subsets for the tags $\{Bamboo, Bishu, Head2Head\}$

by the user. Assigning the threshold can be an especially challenging task. In the following I present another method for automatic annotation free of this weakness.

4.2.2 Multiple Binary Decisions for Communication Annotation

Another simple method to annotate an unseen sequence is to train a classifier for each annotation individually instead of using a probability distribution. This method has the advantage that the threshold will not have to be chosen. For a data set with m tags, first I extract the statistics for each dolphin communication sequence. Then I create m relabelings of the data set. Specifically for every tag, I relabel all dolphin communication sequences annotated with it a positive label and all others with a negative label. The relabeling in our example is:

1. t_1: $positive = \{X_1, X_3, X_4, X_5\}$, $negative = \{X_2\}$
2. t_2: $positive = \{X_1, X_3, X_5\}$, $negative = \{X_2, X_4\}$

3. t_3: $positive = \{X_2, X_4\}$, $negative = \{X_1, X_3, X_5\}$

4. t_4: $positive = \{X_5\}$, $negative = \{X_1, X_2X_3, X_4\}$

Now I can train a single, binary classifier for each annotation. Each classifier returns a positive label if the tag associated with it should be present and a negative label when it should not. The results of the training are m classifiers, one for each annotation.

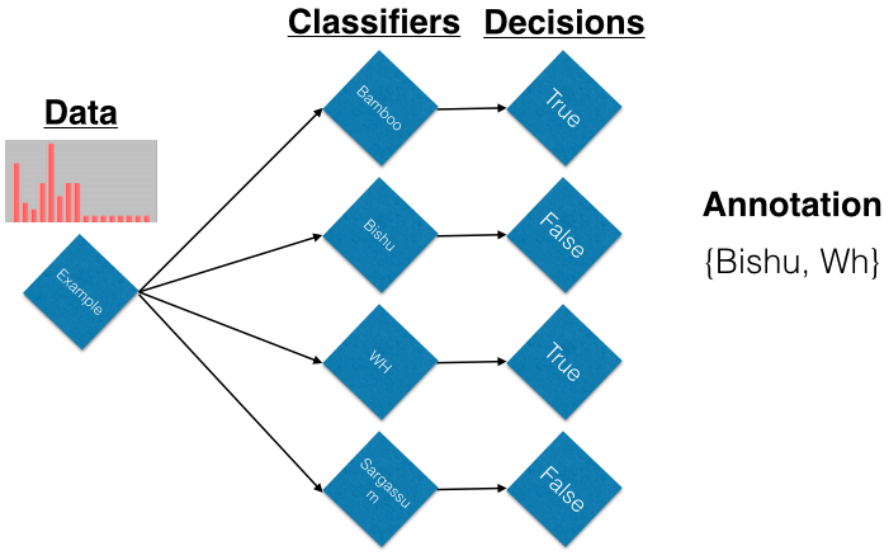


Figure 20: Annotation process: Each example is classified by binary classifiers, each associated with an annotation: *bamboo, bishu, wh, sargassum.* All the classifiers that evaluate positively contribute to the annotation.

I classify each unannotated dolphin communication sequence using all classifiers. The sequence is annotated with all tags for which classifiers returned a positive label. The complete process is shown in Figure 20.

4.3 Comparative Statistics of Dolphin Communication in Context

In the last part of this chapter I will describe how to perform comparative statistics between dolphin communication in different contexts. The statistical models over

pattern occurrences in dolphin communication sequence enable behavior researchers to compare subsets of their data with each other using statistical testing. Today dolphin researchers have no means of statistical testing for audible dolphin communication. Through my discovery algorithms and statistical modeling methods, I enable this novel approach to hypothesis testing in the field of dolphin communication. One example hypothesis is that audible dolphin communication is statistically different during aggressive behavior than during play behavior. In other words the pattern distribution is statistically different in the two behavioral contexts. The biologists can now collect audio files with examples from both contexts and compute the statistics for each of the dolphin communication sequences. Then the algorithm aggregates the statistics for each context individually. The results are two histograms representing the pattern counts, the n-grams, the rules, or a combination of these units. Using the two histograms I perform statistical testing. I run a Pearson's χ^2 test between the two distributions. A χ^2 test evaluates if a set of observations is significantly different from a given distribution. In other words, the null hypothesis is that the observed counts conform to the frequency distribution described by the expected counts. I compute two p-values. The first p-value uses a χ^2 test using the first context's data as the given distribution and the second context's data as the observations. The second test uses the first context's data as the observations and the second context's data as the distribution. The two p-values can then be used by biologists to report numeric evidence for the statistical difference between the two contexts.

CHAPTER V

A USER INTERFACE FOR DATA MINING IN DOLPHIN COMMUNICATION DATABASES

The goal of the thesis is to provide behavior researchers with a tool to discover patterns in dolphin communication and to perform statistical testing. However, such a system has to be able to communicate with the researcher through visualizations, and it has to enable the researchers to run experiments on their own. Running the experiments involves several steps including feature learning, pattern discovery and statistical analysis. In the following, I will describe the signal imager, my user interface for the dolphin communication analysis system. First I will describe my design goals and sketches for my system and then describe the implementation in detail using several use cases.

5.1 Design of the Signal Imager

The signal imager should help researchers to extract features interactively, discover patterns and run statistical analysis. I organize experiments into projects. A project is a folder on disk containing statistical models and audio files needed for the experiments. A project includes a set of feature extractors and a mixture of hidden Markov models. Each audio file in the project is converted to a spectrogram and then converted into the novel feature space. Furthermore, each sequence is decoded using the hidden Markov model resulting in the desired pattern sequences.

I have designed the signal imager to provide easy access to dolphin communication sequences (see Figure 21). The main screen focuses on displaying the patterns found in dolphin communication on top of its spectrogram. The patterns are color coded.

The sidebar on the left enables navigation through dolphin communication sequences, and the visualizations change accordingly. Researchers can use the main interface to browse their data collection and see where the patterns appear in the spectrogram.

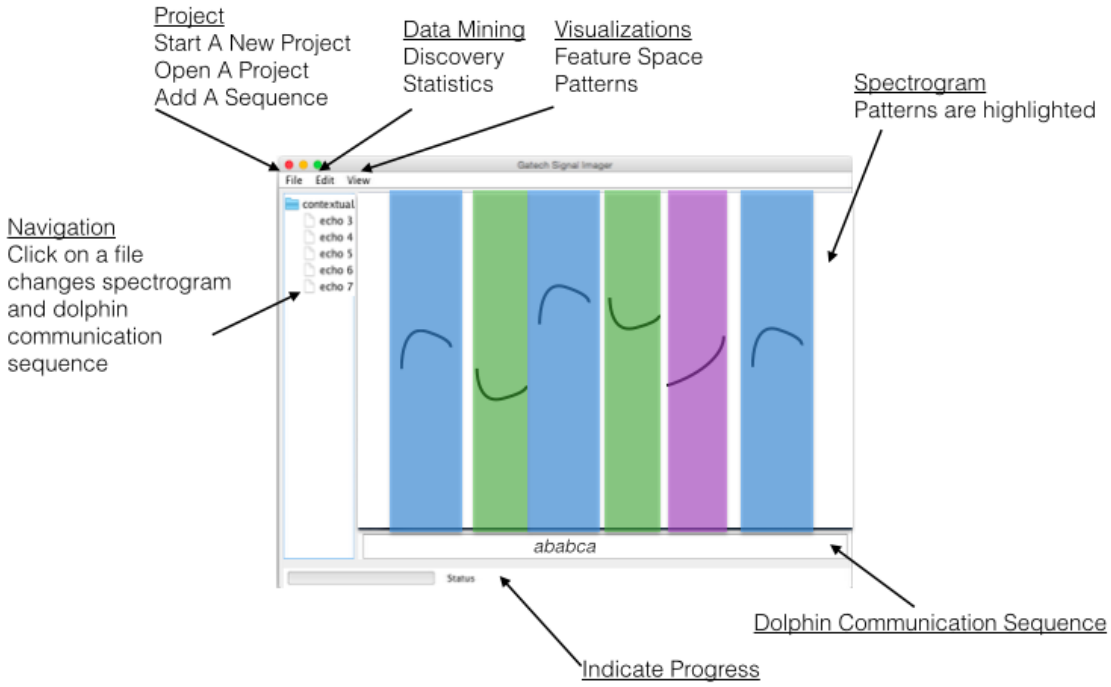


Figure 21: A design sketch for the signal imager.

The system provides basic project management functionality. Users can create a new project, load a project and add sequences to a project. Creating a project involves learning the feature space. The users decide on the sliding window used to compute all spectrograms and the number of codebook entries. Furthermore, the users provide the categorized examples to the feature learning algorithm. The complete process is guided by a project setup wizard. After the creation of a project, users can start adding sequences to it. Adding a sequence to the project will convert a selected audio file into its spectrogram and into the feature space. Both versions are saved in the project folder. Each sequence added to the project will show instantly on the

left side, and users can inspect the spectrogram. The data mining options allow the researchers to build the hidden Markov model and to compute the desired statistics. Once a user creates a hidden Markov model, then all spectrograms show the colored highlights.

5.2 Visualizing Patterns and Statistics

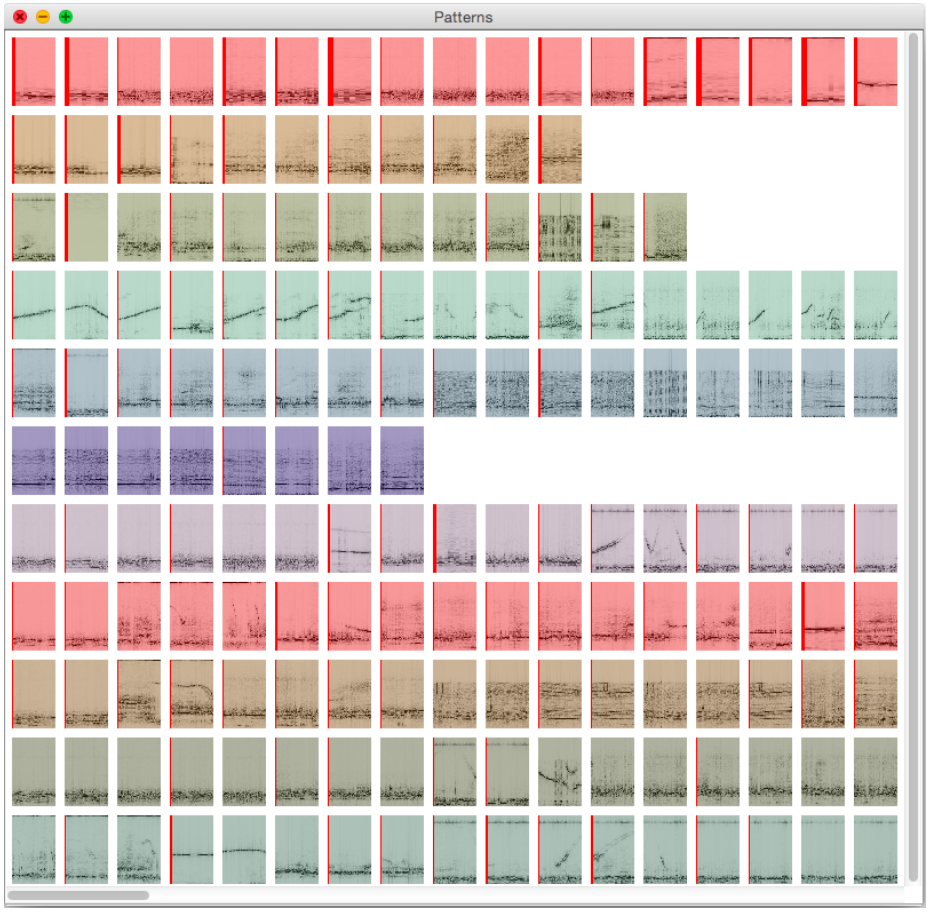


Figure 22: The pattern view of the signal imager. Each row represents one pattern. Each column represents an example of that pattern. All examples are color coded indicating the pattern.

After the user creates the hidden Markov model, the signal imager tool offers several visualizations. The first visualization is the pattern view. The pattern view visualizes all examples of each pattern. The pattern view is a matrix containing all patterns and all examples. A user can inspect all pattern examples in the database

using this view. Each row in the pattern view represents a pattern. Each column in that row represents an example of that pattern. In this way, users can easily examine the goodness of the clusters. If the examples in one row are very similar to the examples in another, then the discovered patterns are not that good. These similar patterns could have been merged during discovery. These errors might occur for feature spaces with too many feature extractors or for hidden Markov model mixtures with too many components. On the other hand, users might discover that the pattern examples are not very similar to each other. In that case, a larger feature space or a larger hidden Markov model might help. The pattern view is shown in Figure 22.

Figure 23: Two histograms. On the bottom histogram, the user interface shows a regular expression revealed by using the interactive interface.

The other important visualization is the statistics view. As mentioned earlier, a dolphin communication sequence can be described as a histogram combining unigrams, n-grams and regular expression matches. The visualizer shows the histogram

of the frequencies for each component. If a user hovers the mouse over one of the bars, the signal imager displays the underlying pattern, n-gram or rule. The visualization allows a quick inspection of the most prominent components in a dolphin communication sequence. An example of an interactive histogram is shown in Figure 23.

5.3 Use Case 1: Finding Patterns in a Database

The first use case explains the process of creating a project and discovering patterns for inspection. Imagine a marine biologist, who acts as the domain expert. She collects video and audio of wild Atlantic spotted dolphins using underwater cameras and hydrophones. After several years of collection, she has access to a large multimedia database containing several hours of footage of the animals' social interactions. Over the years, she has also collected enough insight into dolphin communication to identify several signal categories such as the dolphins' whistles, echolocation and burst pulses. Furthermore, she identified visual patterns in dolphin group behavior such as synchronized swimming during aggression. However, one of her future goals is to identify typical patterns in the dolphin signals and compare the usage of these patterns across several different social contexts. Her problem is that the manual analysis of her audio database is too slow and too subjective to get enough insight into the distribution of patterns to collect evidence for her analysis. My data mining system mostly works on its own. The domain expert's task is to curate data suitable for her experiment and feed it into the tool. Crafting these datasets from her large database is her way of putting domain knowledge into the system.

In order to run her experiment, the domain expert collects a catalog with examples of known signal types. She suspects that mother-calf reunions will show more whistle usage. Furthermore, from her previous research she suspects that dolphins synchronize their burst pulses during aggressive behavior. Her first task is to collect

a small catalog of audio snippets containing several whistles and several burst pulses. She starts by creating a new project, and in the process, tells the program which audio snippets are whistles and which are burst pulses. Using that knowledge, the signal imager starts to learn a feature space that captures the dynamics of whistles and burst pulses by analyzing small patches extracted from the spectrograms of the given examples. For example, for whistles, the program will learn several patches showing characteristic up and down sweeps (see Figure 24). The program uses these patches to build a new feature space in which dolphin signals become easy to detect and compare. The feature space captures the presence in the spectrogram of each of the patches over time.

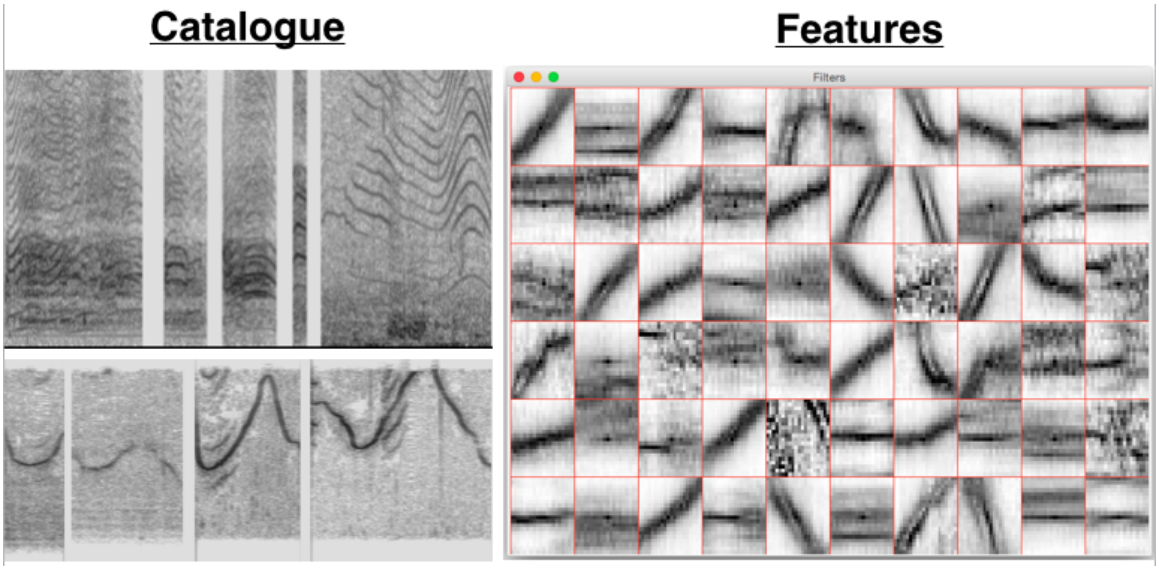


Figure 24: Left: A small excerpt of the curated examples collected by the domain expert. Right: The resulting patches learned by the program.

After the domain expert estimates the feature space, she has a method to capture the dynamics of dolphin communication as provided by her examples. Next she collects several examples of dolphin communication during aggressive behavior and mother-calf reunions. From her video annotations, she can access her database in a timely manner and extract the audio from the associated video files in regions where she observed the desired behavior. She adds all the examples to the newly

created project. The signal imager converts her audio files into two representations: a spectrogram representation for visualization purposes and the feature space using the features capturing the dolphin signal dynamics for pattern discovery. The user interface for such a project is shown in Figure 25. On the left side, the domain expert can choose one of the files she added, and it will be displayed in the center.

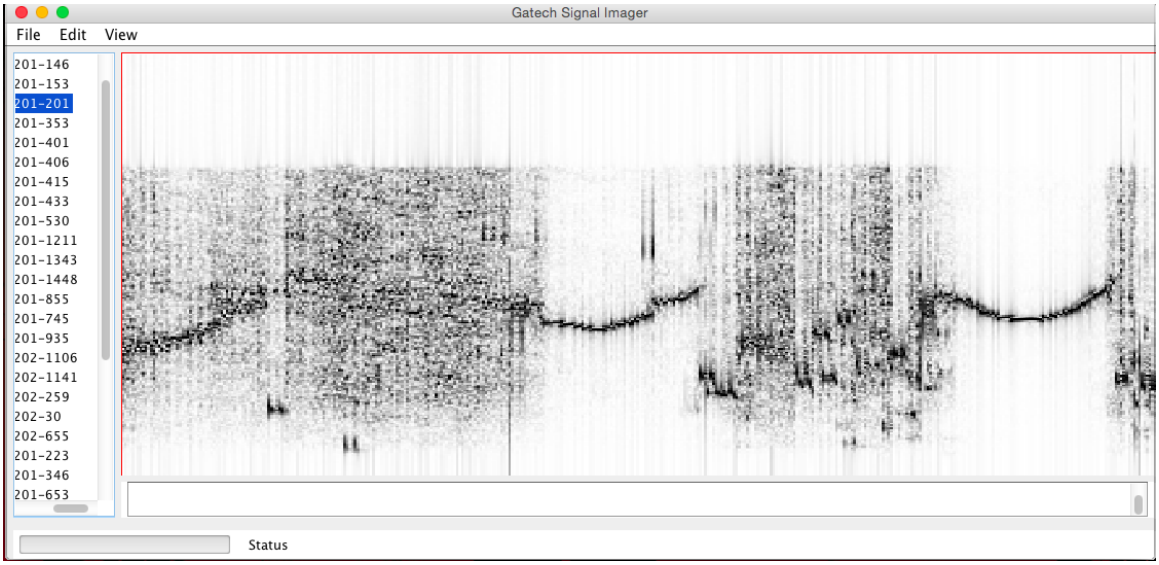


Figure 25: The user interface with dolphin communication examples added. It shows all the files on the left and displays the spectrogram for selected files on the right.

The next task is to segment each of the files, or in other words, find common patterns in the files. The segmentation is performed automatically on a button press. The program extracts regions of dolphin communication in the new feature space and clusters these regions. The number of cluster components is learned automatically. After the tool finishes the discovery work, it displays the patterns by coloring regions in the spectrogram. Furthermore, the tool displays strings at the bottom of the screen indicating which patterns are found (see Figure 26). The domain expert then inspects the files and the discovered patterns.

Now that all of the dolphin signaling examples are in a discrete representation, the domain expert can start modeling statistical patterns in the program. The user interface provides an option to compute statistics from the discrete representation.

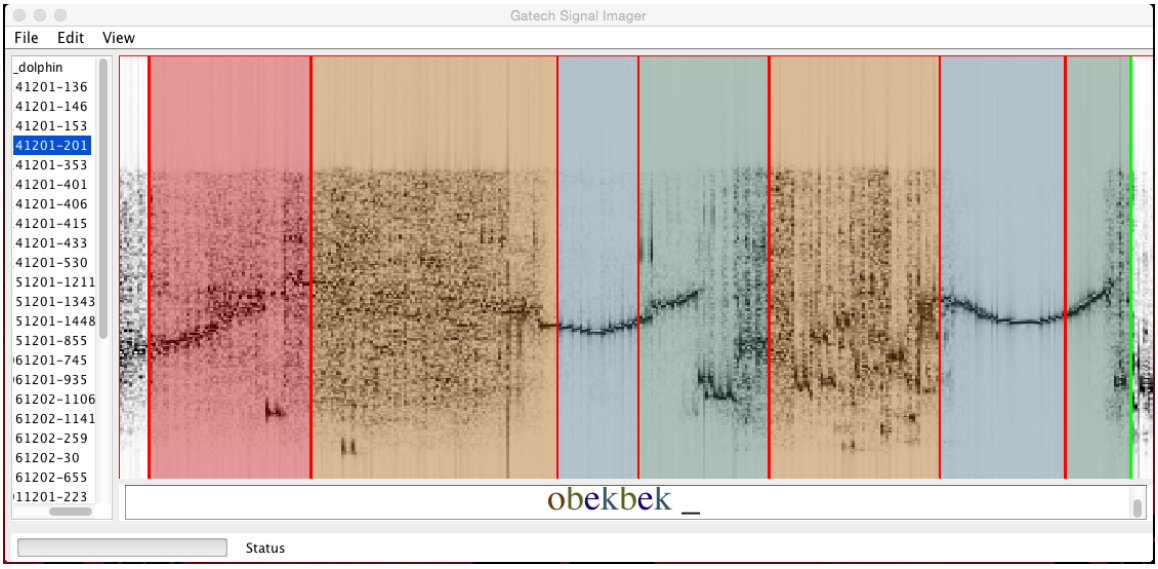


Figure 26: The user interface with color highlighted segments.

5.4 Use Case 2: Comparative Context Analysis

Using the newly created project with the added sequences and discovered patterns, the domain expert proceeds to calculate statistics and to run a comparative analysis. The signal imager offers a new window to compute statistics (see Figure 27).

In the statistics view, the domain expert can choose from several statistics including the bag-of-words model, the n-gram model and the regular expression (Figure 27, top left). Checking each box will enable the model. The parameters can be input next to the model name. Afterwards, the domain expert is ready to run a comparative statistic. The statistics screen is located at the bottom of the window. The goal is to select two subsets of files from the project and run a χ^2 test between the aggregated histograms of each subset. The user interface shows two columns. Both columns list all files. The user selects the first subset in the left column and the second subset in the right column. In our case, the researcher selects all data including play behavior in the left column and all data containing mother-calf reunions in the right column. Pressing the run button will trigger the signal imager to run the test. The program will calculate all selected statistics for all selected files. Afterwards, all histograms

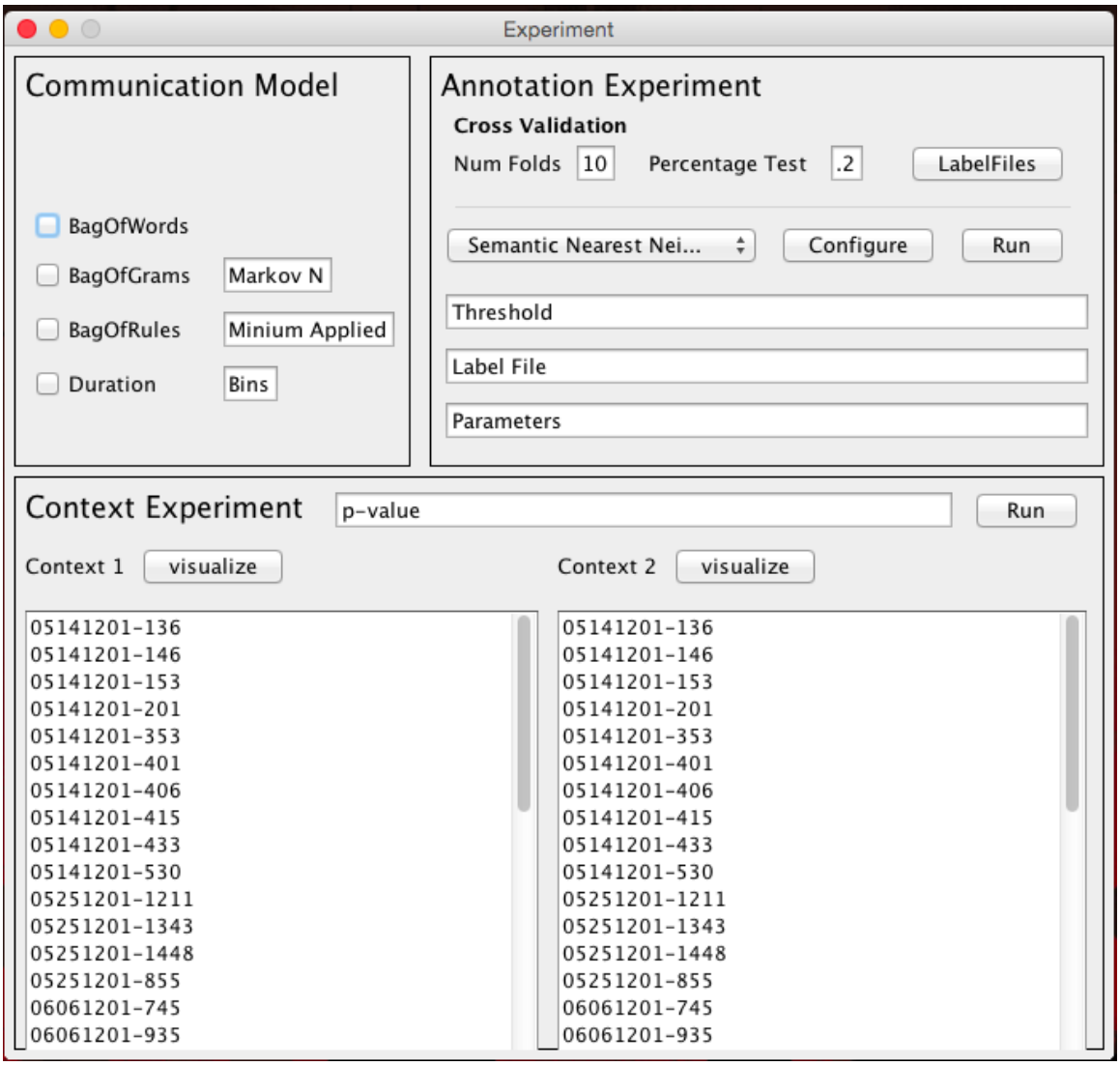


Figure 27: The statistics view includes the feature selection on the top left, the annotation experiments on the top right and the statistical testing at the bottom.

on the left column are aggregated by summing all counts in each bin. The same aggregation is applied to the right column. Then the signal imager calculates the χ^2 test between the two histograms and writes the result in the $p - value$ text field. The behavior researcher can now evaluate if there is a statistical difference between audible communication during play behavior and communication during aggressive behavior. The visualize button above each column opens the interactive histogram view for further inspection.

5.5 Use Case 3: Annotation of Novel Dolphin Communication

In the last experiment, the dolphin researcher wants to compare the performance of several parameters. One example case might be that the researcher chose the fast algorithm to discover patterns in the audio files, and in another project with the same data she used the exact algorithm. Now she wants to compare the performance of both algorithms numerically. The signal imager provides the option to run an annotation experiment. Provided with an annotation file for the sequences in the project, the signal imager can train a classifier and compute the confusion matrix and accuracy over several folds of the dataset. If she performs the annotation experiment in both projects, she can compare the accuracy, precision and recall of several classifiers for the project with the exact discovery and the project with the fast discovery. The same experiments can help her to decide which statistics work best. An example comparison is to compare bag-of-words only with bag-of-words combined with the rule model.

If the domain expert is not happy with the default options of the tool, her pattern recognition colleagues can easily help her with the classifier selection using the well-known Weka interface, whose toolkit is included in my program (see Figure 28). After the cross-validation experiments, the program shows the confusion matrix with precision and recall as well as the annotation accuracy. The program also visualizes the true positives for each annotation. For example, if the complete system can only annotate dolphins' names, the domain expert might conclude that the system only works for whistles. However, if higher level annotations such as "head-to-head swimming" are predicted well, she can form hypotheses about the audio patterns during aggression. Figure 29 shows the result view for the annotation experiment. The confusion matrix and the precision, recall and accuracy of the annotatoin are shown at the bottom of the left window. The top of the left window shows the status

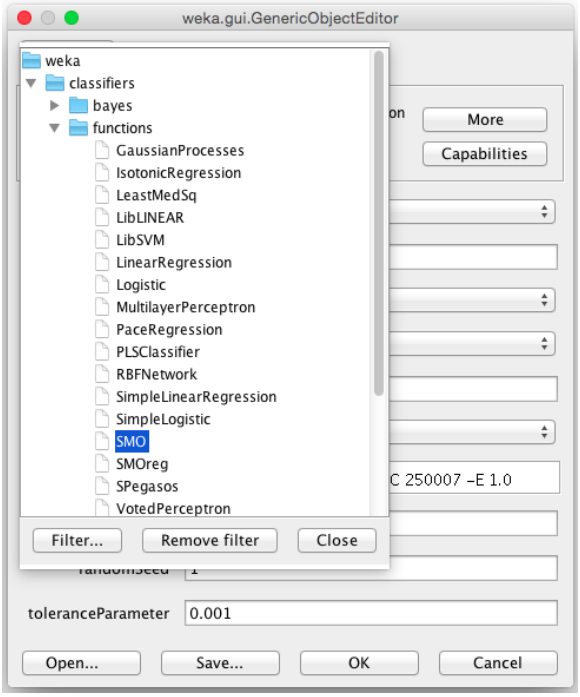


Figure 28: The standard classifier selection from the Weka interface.

of a cross-validation experiment. The right window shows a histogram. Each bar represents the number of true positives for each annotation.

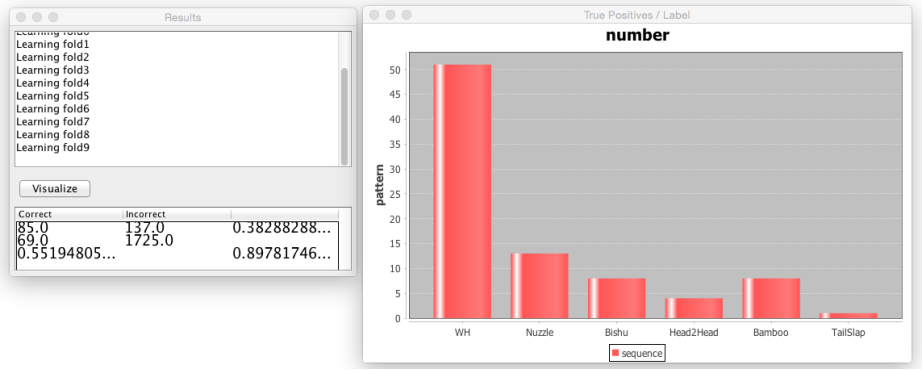


Figure 29: The annotation results view.

Using the signal imager, the domain expert can discover patterns in her field data. Furthermore, the signal imager allows her to run statistics and annotation experiments on her own and visualize the results. In the following chapter, I will present my experimental evaluation of the program including a user study with a

biologist.

CHAPTER VI

EVALUATION

In the previous chapters, I presented how the signal imager processes a collection of audio files. The program learns a feature space and transforms the spectrogram of each audio file into a new feature space. Then the discovery algorithm finds patterns in this novel feature space. In the end the signal imager builds communication models that allow statistical testing of communication in different contexts and the annotation of dolphin communication. In this chapter, I present my evaluation of the signal imager program and its algorithm. In the first two experiments, I evaluate the annotation performance and the statistical testing performance using ground truth datasets. In the last experiment I evaluate the quality of the pattern discovery with a domain expert. In the last section I will highlight common errors that I observed during my experiments. The experiments will show that my proposed system is capable of hypothesis testing and generation. Furthermore, the last experiment will show that the discovery results are indeed useful and interpretable by domain experts.

6.1 Signal Imager Experiments

The following three experiments evaluate the signal imager's performance in several conditions. In previous chapters, I proposed several different algorithm and modeling options for pattern discovery and statistical modeling.

In the first experiment, I show how the annotation performance of the signal imager changes for several communication models and for different discovery algorithms. For example, one question is how a bag-of-words model compares to a bag-of-words model augmented with an n-gram model. Another question is how the approximate discovery algorithm compares to the exact discovery algorithm. The hypothesis is

that an appropriate communication model and better patterns will increase the annotation accuracy.

In the second experiment, I evaluate the statistical testing performance. The hypothesis is that there is a statistical difference between communication in different behavioral contexts, such as play and aggression. In the experiment, I use a dataset with five different behavior contexts and show that the system discovers the appropriate statistical differences. Furthermore, I show that the discovered patterns have a direct effect on the testing performance and how better patterns can be discovered by adding unlabeled data.

In the last experiment, I evaluate the quality of the pattern discovery with a domain expert. The biologist inspects the patterns in several conditions and gives an expert opinion. During the study, the domain expert will rate the pattern discovery results for the approximate algorithm and the exact algorithm. The experiment shows the perceived performance of the system.

6.1.1 Automated Behavior Tagging

In this experiment, I will evaluate how the annotation accuracy changes for different communication models and the two pattern discovery algorithms.

The annotation experiment is designed to answer multiple questions:

1. Can the signal imager annotate unseen dolphin communication sequences?

2. What are the best statistics to annotate dolphin communication sequences?

3. Is there a difference between the exact and approximate algorithm?

I use two datasets: the categories dataset and the annotated dataset. The categories dataset consists of short audio snippets. Each snippet is approximately one second or less and is categorized as a whistle, a burst pulse, echolocation or simply noise. There are 3 noise files, 78 whistles and 15 burst pulse examples. The noise

and burst pulse files are a second in length or shorter. The noise files combined are about two minutes in length.

The annotated dataset contains 67 audio files representing annotated dolphin communication. I extracted the audio files from audiovisual recordings made in 2012. I use video annotations to extract the 67 audio files and use the behavior tags as annotations. All audio files in both datasets show a sampling rate of $44.1kHz$. I calculate all spectrograms using a sliding window of 512 samples with a 102-sample overlap. Furthermore, I apply a Hanning window to each of the sliding windows. The feature transformations are learned from $20x20$ patches and contain 60 feature extractors. I learn the feature extractors from the category data set. I use the 67 annotated audio files from 2012 for this experiment.

After I convert all 67 audio files into a 60-dimensional feature space, I use the resulting time series to compute two sets of dolphin communication sequences. The first set is constructed by discovering patterns in the dataset using the exact algorithm. The second set is constructed by discovering patterns in the dataset using the approximate algorithm. For the initial segmentation into signal and noise, I use a random forest with 10 trees. I use a three-state, left-to-right hidden Markov model for each pattern in both cases. Furthermore, the Gaussian mixture model for the noise state in the final hidden Markov model mixture has three components. During the exact algorithm, the final hidden Markov model mixture is initialized with 20 clusters found using hierarchical clustering under dynamic time warping distance. The approximate algorithm initializes the hidden Markov model mixture from the data of all hash bins with more than 10 sequences.

My evaluation is based on several different classifiers and statistics combinations. The classifiers I use are a support vector machine, the semantic k-nearest neighbor with a probability threshold of 0.005, a naive Bayes classifier and a random forest. The support vector machine (SVM) uses a radial basis function kernel with the standard

Weka parameters. The semantic k-nearest neighbor (KNN) algorithm extracts three neighbors for each tag. The naive Bayes classifier (NB) has no parameters to choose. Finally, the random forest (RF) contains 10 trees. Each tree is restricted to a depth of seven and selects from a set of five random features for each decision. All classifiers except the semantic k-nearest neighbor algorithm perform a binary decision for each behavior tag.

In the first experiment, I use several sets of statistics as features for the annotation and observe the accuracy for all classifiers. The statistics are bag-of-words (BOW), n-grams (BOG), regular expressions (BOR), bag-of-words combined with regular expressions, bag-of-words combined with n-grams, regular expressions combined with n-grams and all three models combined. I extract all regular expressions and use the ones that are used more than eight times for the statistics. I choose to use bi-grams for the statistics. For this experiment, I use the dataset resulting from the exact algorithm. I split the dataset into 80% training data and 20% test data using 10 folds of Monte Carlo cross-validation. I measure the number of correct annotations (true positive), in each condition, the correctly omitted annotations (true negative) as well as the falsely omitted annotations (false negative) and false annotations (false positive). I calculate the accuracy for each condition from these counts:

$$accuracy = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}} \tag{30}$$

The results of the experiment are shown in Table 2.

I observe the lowest accuracy of 78% for the bag-of-words condition with the semantic k-nearest neighbor algorithm. The highest accuracy is 91% and is achieved using a random forest and using the combined statistics of bag-of-words and the regular expressions. The same condition also shows the highest average accuracy across classifiers. Performing a t-test between all conditions is inconclusive. There are only minor statistical differences before Bonferroni corrections and none after. One reason

Table 2: The results of the annotation experiments for several statistics.

	BOW	BOG	BOR	BOW, BOG	BOW, BOR	BOG, BOR	BOW, BOG, BOR
SVM	79%	89%	87%	89%	88%	87%	87%
KNN	78%	80%	80%	79%	80%	80%	80%
NB	80%	80%	88%	80%	88%	88%	87%
RF	88%	90%	90%	89%	91%	90%	89%
AVG	81%	84%	86%	84%	87%	85%	85%

might be that there is not enough data for cross-validation to show these differences. In order to establish a baseline, I build a classifier that rejects all annotations. In other words, there are no true positives and the accuracy of the baseline classifier is based on true negatives alone. Some of the accuracies are equivalent to the baseline accuracy which is 79%. In these cases, it is not clear if the classifier predicts any annotations correctly or not. To get a better idea about the breakdown of the accuracy into true positives and true negatives, I repeat the experiment for the bag-of-words combined with the rules condition. I use the same classifiers and compute the precision and recall for each. Precision can be regarded as the percentage of correctly annotated data from all annotations made. Recall is the percentage of correct annotations from all possible correct annotations. Precision and recall are defined as follows:

$$precision = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{31}$$

$$recall = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{32}$$

The precision and recall for the four classifiers are shown in Table 4. According to the results, it seems that the discovered patterns provide enough discriminative power to distinguish between the biologist's annotations, and the annotation algorithm predicts several annotations correctly. I also calculate the precision and recall for each annotation individually.

Table 3 shows how often each annotation is used in the dataset. Furthermore, the table shows the precision and recall for each label individually computed using all four classifiers. It is not surprising that often-used annotations are the highest percentage correct. As one can see, the algorithm is able to annotate visual identifiers well and even shows some predictive power towards visual behavior. These results indicate that the patterns found in the dataset might represent semantically meaningful signals such as signature whistles or other signals that are predictive of visual behavior. A detailed inspection of the precisions and recalls indicates that the algorithm is capable of predicting a dolphin's visual identifier. One possibility is that the algorithm uses patterns that represent signature whistles. The algorithm is also capable of predicting the head-to-head annotation that is observed during aggressive behavior. One possibility is that the algorithm uses patterns that represent a specific burst pulse sound called a synchronized squawk. That sound is often associated with aggressive behavior, too.

In the second experiment, I compare the approximate algorithm with the exact algorithm. Since the bag-of-words statistics combined with the regular expressions worked best in the first experiment, I use these statistics for the second experiment. I extract the statistics for the dolphin communication sequences generated from the approximate and exact discovery algorithms and compare the two conditions with each classifier. The results of the second annotation experiment are shown in Table 5. In the second experiment, the exact algorithm performs better than the approximate algorithm. Calculating the t-statistic reveals that the exact algorithm is in fact weakly significantly better than the approximate algorithm, with a $p - value$ of 0.03. The results in these experiments indicate that the patterns found in the audio stream provide enough discriminative ability to annotate audio data with annotations created from visual observations. The precision in the experiments is high enough to enable efficient search in a database of audible dolphin signals in the future work. In the next

experiment, I show that the patterns' discriminative power is sufficient to distinguish between higher behavioral contexts using statistical testing.

6.1.2 Comparing Communication Among Different Dolphin Behavioral Contexts

The second series of experiments evaluates the statistical testing function of the signal imager. I use another dataset to run comparative statistics among different behavioral contexts. The audio files provided by the domain experts are annotated with the behavior contexts: play behavior, foraging behavior, aggressive behavior and mother-calf reunions. In total, the dataset contains 25 audio files: 7 files showing aggressive behavior, 5 files with foraging behavior, 6 files with play behavior and 7 files with data from mother-calf reunions. The domain expert picked these contexts since the expert community has agreed that communication in these contexts is different. For example, foraging behavior includes mainly echolocation; aggressive behavior includes mainly burst pulses; play behavior includes whistles; and mother-calf reunions include signature whistles. When comparing these contexts to each other, they should all show a significant difference in the pattern distribution. Furthermore, when comparing a context to itself, it should not show a significant difference.

In particular I want to investigate the following questions:

1. Is the signal imager able to confirm the differences between the four contexts?

2. Does adding more, unlabeled data lead to a better pattern estimate?

For the following experiments I use two datasets. The first dataset contains the four-context dataset. The second dataset contains the four-context dataset combined with the patterns from the 2012 field season. For both datasets, I convert all audio files into the 60-dimensional feature space using the feature extractors learned from the category dataset. For both datasets I apply the exact algorithm with the same parameters as in the annotation experiment, resulting in the following two conditions:

1. Exact discovery in four contexts only.

2. Exact discovery in four contexts combined with the patterns from the 2012 field season.

I run comparative statistics in each condition and perform comparisons between two contexts using a χ^2 test on the histogram statistics extracted from each dolphin communication sequence. From the annotation experiment, it seems the regular expressions combined with the bag-of-words model show the best performance during annotation. Therefore, I decide to use these statistics in these experiments, too. I proceed with the following testing procedure. When comparing data from context c_1 and context c_2:

1. Estimate a distribution c_1, and test if c_2 is from that distribution.

2. Estimate a distribution c_2, and test if c_1 is from that distribution.

The method returns two $p-values$, one for each case. These $p-values$ are used to indicate the significant difference between the communication in each context. At this point in the data mining pipeline, the audible communication is described by a distribution estimated from the discovered patterns. In other words, the $p-values$ indicate significant differences in communication among different contexts indirectly through the estimated pattern distributions. In my experiments, I use a 0.95 confidence interval. That means I reject a hypothesis if the $p-value$ is greater than 0.05. For example, if the $p-value$ between the contexts play behavior and mother-calf reunion is smaller than 0.05, then the difference in the communication is significant. When the $p-value$ is larger than 0.05, then the difference is not significant. Furthermore, when testing multiple contexts, I apply the Bonferroni correction. The Bonferroni correction accounts for the familywise error rate, which means that the correction accounts for a type I testing error (i.e., false positive) when performing

multiple comparisons. The Bonferroni correction is the most conservative of correction methods. Other options include resampling or permutation testing. However, these methods require more testing data than are available in my experiment.

When comparing data from a context to itself, I split the data into two equal subsets and run the tests between the two. I repeat the testing process 10 times and average the resulting $p - values$. The results of the multiple tests for the first condition are shown in Table 6. Each rule has to be used more than eight times to contribute to the statistic.

The values on the diagonal of the table show the $p - values$ for the comparison of each context to itself. As assumed, the values show no significant difference. All the other entries in the table show the $p - value$ for the comparison of two different contexts. These values should show a significant difference between the contexts. However, I observe one error in the results. It seems that foraging behavior and play behavior are not significantly different. In other words, the discovered patterns indicate that the communication in both contexts is not significantly different. However, from the domain experts, we know that the communication during foraging behavior is characterized by echolocation and that the communication during play is often characterized by whistles.

In the second experiment, I combine data from the 2012 dataset with the four-context dataset. My hypothesis is that a larger set of data will stabilize the pattern discovery and in turn provide cleaner statistics. In this experiment, I run the pattern discovery and the rule discovery on the combined dataset. I then perform the same statistical testing as in the first experiment. The results of the second experiment are shown in Table 7.

As one can see, the combined dataset shows higher $p - values$ for the diagonal and lower $p - values$ in every other cell. Furthermore, the errors in the combined condition only occur after correction for multiple tests. The first experiment shows

an average $p-value$ of 0.45 on the diagonal and an average $p-value$ of 0.1 in every other cell. The exact algorithm shows an average $p-value$ of 0.78 on the diagonal and a $p-value$ of $2.88e^{-3}$ in every other field. In other words the combined condition follows our expectation more strongly. Comparing the context to itself should result in a highly non-significant estimate. The expectation is that communication in playful behavior is similar across instances. Furthermore, I expect every comparison between contexts to be very different, since the biologist picked the context examples to be different in behavior and communication. In other words, using the larger dataset follows our expectation, while the context dataset alone results in an error.

6.1.3 Qualitative Analysis of Pattern Discovery

In this section, I evaluate the pattern discovery results with the domain expert. As indicated by the previous experiments, there is a difference between the fast and the exact algorithm and between the small and large dataset. For the evaluation, I use the pattern visualization of the signal imager. I discover patterns in the following four conditions:

1. Small dataset with the exact discovery.
2. Small dataset with the approximate discovery.
3. Large dataset with the exact discovery.
4. Large dataset with the approximate discovery.

In the experiment, the domain expert does not know how the patterns are found and which dataset is used. For the purpose of the experiment, the conditions are labeled A, B, C and D. The domain expert opens the precomputed patterns in the signal imager and evaluates them in a think-aloud protocol. She inspects every pattern and its examples visually. Then, the discussion with the domain expert is guided by the following questions:

1. Does using method A result in clean patterns?

2. Does using method A result in distinct patterns?

Both questions are answered on a Likert scale from one to seven using the questionnaire shown in Appendix A. A result of one indicates a strong disagreement, and a result of seven indicates a strong agreement. The first question aims to find the perceived intra-cluster distance. In other words, how similar to each other are the examples assigned to a pattern? The second question aims to find the perceived inter-cluster distance. That is, how distinct from each other are patterns? A high intra-cluster distance would indicate that a lot of examples are assigned to the wrong pattern. A low inter-cluster distance indicates that two patterns that the domain expert would rate the same are split into multiples by the discovery algorithm.

In the first two sessions, the domain expert is presented with the small dataset. The first dataset inspected with the domain expert is the result of the approximate discovery algorithm run on the four-context dataset only. An excerpt of the pattern view is shown in Figure 30. The domain expert notes that the first and the fourth rows seem to gather many different patterns. Furthermore, from the eleven patterns presented, the domain expert would use six or seven. The patterns seem to capture different frequency-modulated burst pulses.

In the second session, the expert inspects the patterns from the exact algorithm run on the small dataset. In the expert's opinion, the exact algorithm on the small dataset seems to perform the worst. There are not enough patterns, and the patterns displayed seem to be quite unclean. The signal imager view for the patterns is shown in Figure 31. The algorithm finds five patterns. In both versions, the domain expert notes that some patterns are very clean while others are either unclean or redundant. However, she notes that it seems easier to relabel patterns when there are too many instead of analyzing unclean patterns.

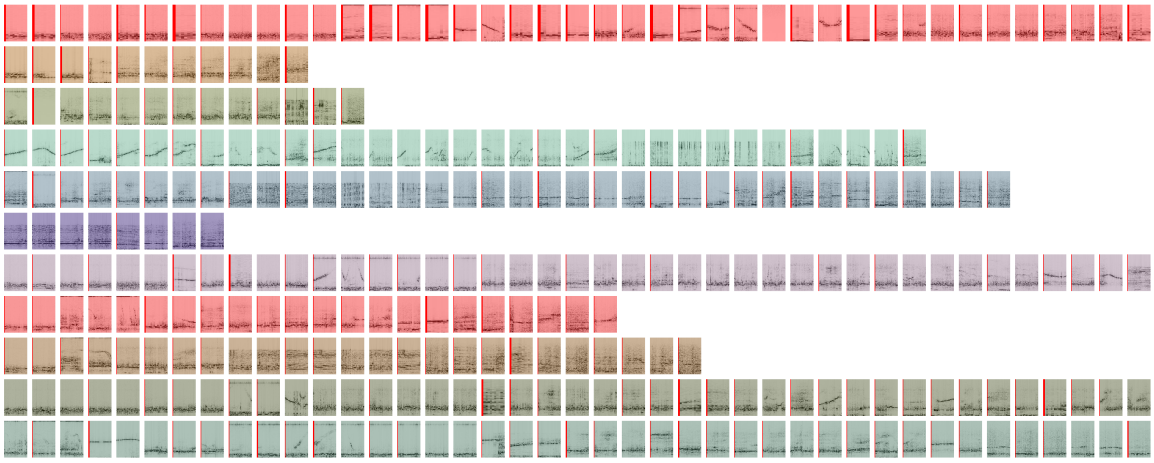


Figure 30: The pattern view in the signal imager for the approximate discovery in the small dataset.

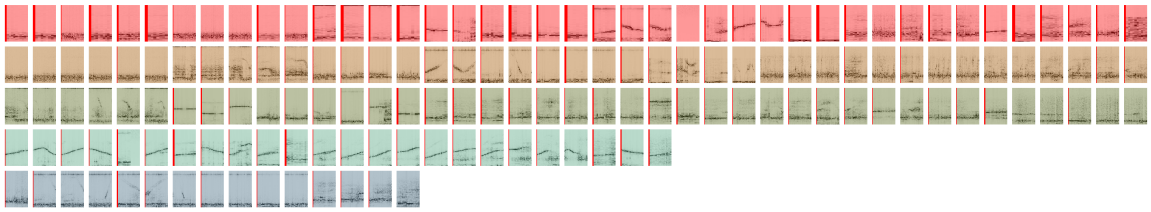


Figure 31: The pattern view in the signal imager for the exact discovery in the small dataset.

The third session inspects the discovery results of the approximate algorithm on the large dataset (see Figure 32). On visual inspection, the domain expert's opinion is that the patterns are very clean. However, she notes that the algorithm seems to be "picking up on whistles," and she thinks that the patterns found in this version are not very distinct from each other. She also notes that she likes this version more than the two previous ones. She also commented that this version seems more promising since the patterns start to look like units of dolphin communication. This result is not surprising since the dataset is larger, and in turn, discovering patterns is easier since there are more examples available to the algorithm.

In the last session, the domain expert inspects the patterns from the exact algorithm on the large dataset (see Figure 33). The expert notes that the patterns are the best so far, since they are distinct from each other and the patterns are clean

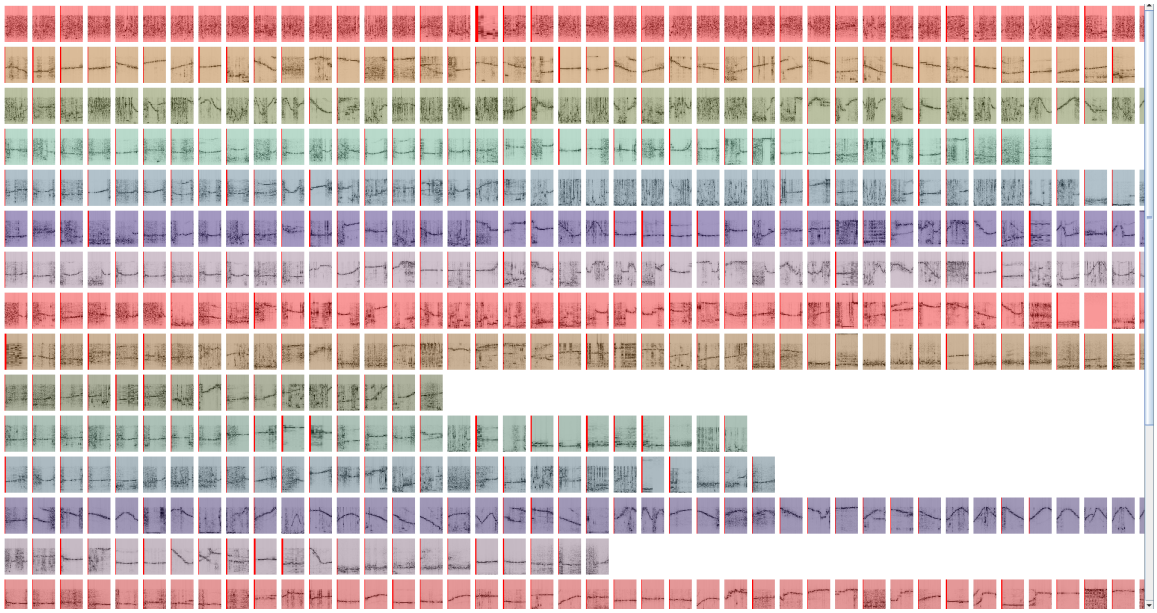


Figure 32: The pattern view in the signal imager for the approximate discovery in the large dataset.

enough.

The results of the Likert scale are shown in Table 8. The first observation is that the domain expert rates the patterns in the larger datasets higher with respect to the inter-cluster distance. The best rating is achieved for the large dataset with the exact algorithm. The Likert scale ratings reflect the results of the discussion.

6.2 Common Errors in Dolphin Communication Mining

In the following, I will describe common error sources that occur during my experiments. First, I will describe errors during whistle tracing. If a whistle is not traced correctly, it might lead to undesired non-dolphin signals in the feature extraction. Second, I will point to common signal detection errors. I use the signal detection results to initialize the patterns. Errors in signal detection will affect the pattern discovery process. The last section describes clustering errors found during discussions with the domain expert.

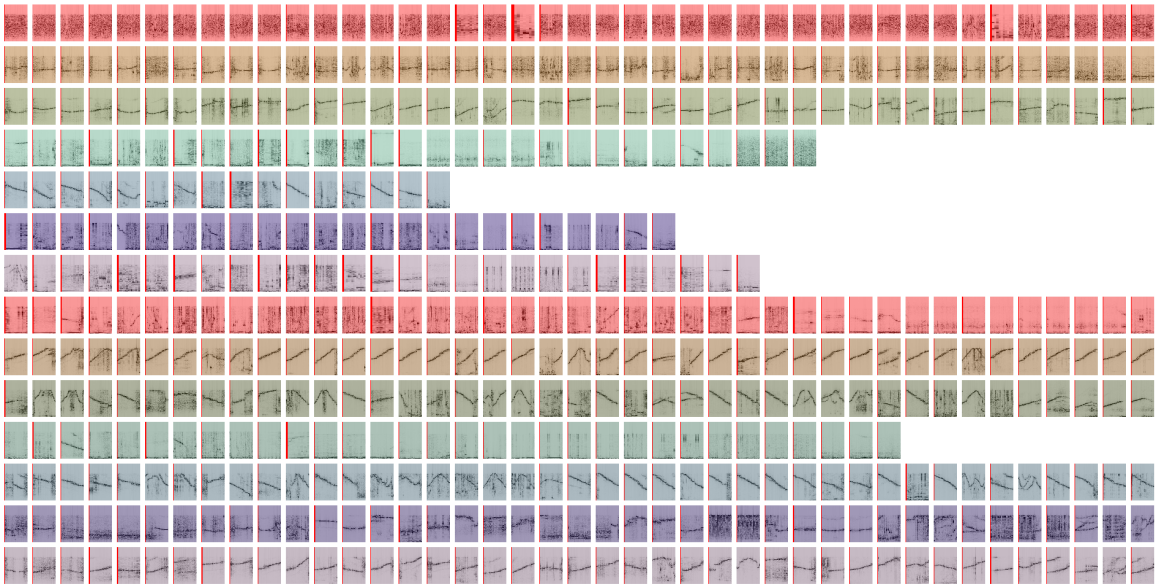


Figure 33: The pattern view in the signal imager for the exact discovery in the large dataset.

6.2.1 Whistle Tracer Errors

In a small set of experiments, I found multiple examples of noise leading to failures during tracing. Often these errors happen when the whistle becomes nearly indistinguishable in the spectrogram, and the signal's dynamic probability is much lower than the magnitude of another frequency. The obvious example is that low-frequency boat noise is louder than the dolphin whistle over a longer period of time. Another example is when a harmonic is much louder than the actual whistle. Some example traces with errors are shown in Figure 34.

6.2.2 Segmentation Errors

In another experiment, I evaluate how effective the feature space and the signal detector are. I use the whistle catalog and the noise examples as training examples. I transform each example into the novel feature space. I then build a random forest with 10 trees, each pruned at a depth of seven. I continue by classifying each example using the trained trees. Using a 10-fold cross-validation with a random split into 90% training data and 10% testing data I observe that 95% of instances are classified

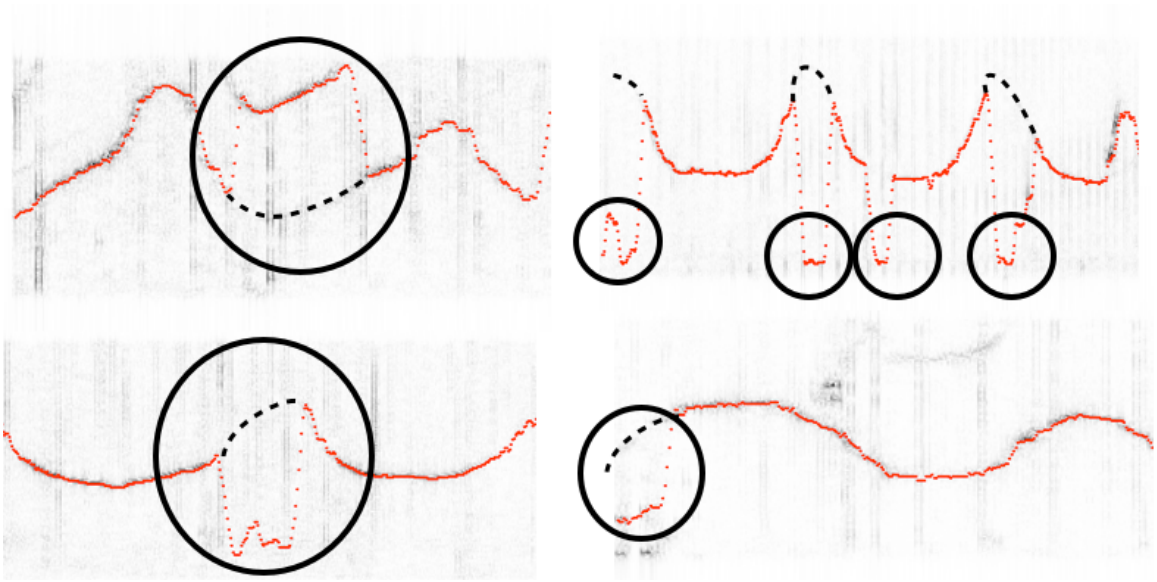


Figure 34: Some failures observed during tracing. Circles indicate error regions. The dashed lines follow the hypothesized actual trace.

correctly. Inspecting the segmentation algorithm for the communication sequences, I observe common errors such as framing, false positives and false negatives. Some examples can be seen in Figure 35.

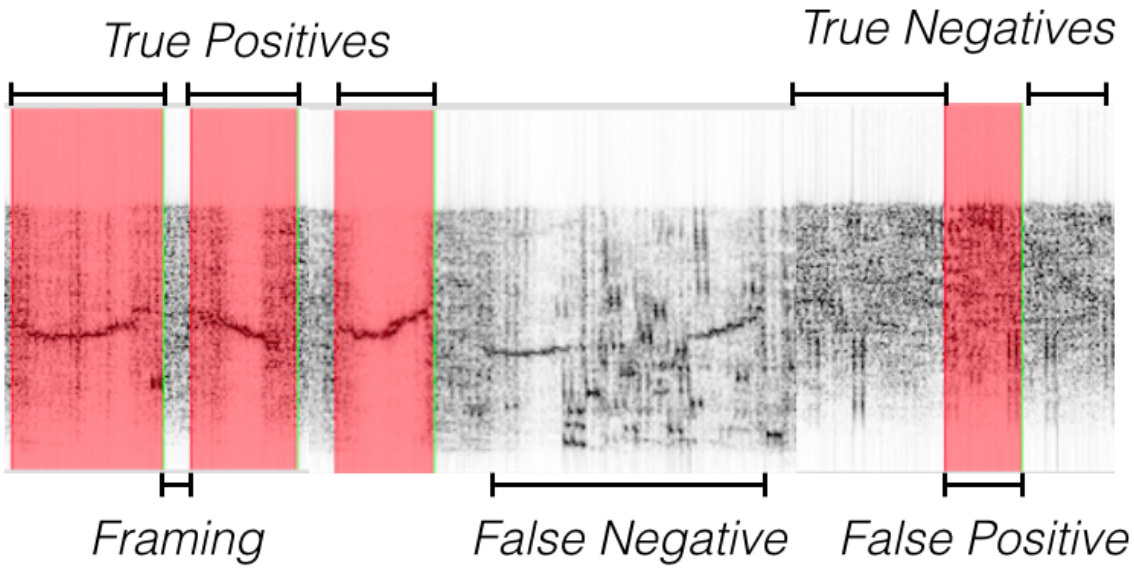


Figure 35: A selection of common tracing errors.

Table 3: A list of all the annotations found in the 2012 dataset with for four classifiers.Each field includes precision / recall.

Label	Count	SVM	KNN	NB	RF
WH	41	0.50 / 0.67	0.60 / 1.00	0.43 / 0.50	0.40 / 1.00
Littleprawn	9	0.00 / 0.00	0.83 / 0.56	1.00 / 0.22	0.00 / 0.00
Bamboo	9	0.40 / 0.29	0.22 / 0.67	0.33 / 0.75	0.29 / 0.50
Nuzzle	9	0.78 / 0.88	0.22 / 0.67	0.33 / 0.75	0.67 / 0.57
Littlegash	8	0.00 / 0.00	0.83 / 0.56	1.00 / 0.22	0.00 / 0.00
Bishu	8	0.40 / 0.29	0.22 / 0.67	0.33 / 0.75	0.50 / 0.50
Nautilus	6	0.00 / 0.00	0.33 / 0.75	0.00 / 0.00	0.00 / 0.00
Play	6	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Head2Head	6	0.67 / 0.22	0.43 / 0.75	0.00 / 0.00	1.00 / 0.17
Ginger	5	0.33 / 0.20	0.43 / 0.75	1.00 / 0.60	0.25 / 0.50
Gelato	5	0.33 / 0.20	0.43 / 0.75	1.00 / 0.60	0.25 / 0.50
SW	4	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
OpenMouth	4	0.00 / 0.00	0.00 / 0.00	0.20 / 0.20	0.00 / 0.00
ECH	4	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Malachite	4	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Mugsy	3	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Cobalt	3	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Sync	3	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
TailSlap	3	0.00 / 0.00	0.20 / 0.33	0.00 / 0.00	0.00 / 0.00
Naia	2	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Nematocyst	2	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Val	2	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Bottom	2	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Sargassum	2	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Chase	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Calve	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Fecal	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Fish	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Littlegash and Littleprawn	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Nautilis	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Flexion	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Venus	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Fused	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00
Discipline	1	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00

Table 4: Precision and recall in the bag-of-words combined with rules condition for the four classifiers.

Classifier	Precision	Recall
SVM	0.66	0.31
KNN	0.26	0.44
NB	0.35	0.43
RF	0.63	0.26

Table 5: The results of the annotation experiments using statistics from the approximate algorithm and the exact algorithm.

	Approximate	Exact
SVM	87%	88%
KNN	78%	80%
NB	84%	88%
RF	90%	91%
AVG	85%	87%

Table 6: The p-values for the statistical testing experiment using the small dataset and the exact algorithm. Significant p-values after correction are shown in green. Non-significant values are shown in blue.

	aggression	play	foraging	reunion
aggression	0.26	$4.8e^{-11}$	$< e^{-14}$	$3.2e^{-8}$
play	$< e^{-14}$	0.23	0.99	$8.8e^{-13}$
foraging	$< e^{-14}$	0.27	0.91	$< e^{-14}$
reunion	$3.0e^{-5}$	$< e^{-14}$	$< e^{-14}$	0.26

.

Table 7: The p-values for the statistical testing experiment using the combined dataset and the exact algorithm. Significant p-values after correction are shown in green. Non-significant values are shown in blue. Values that are non-significant after correction are shown in yellow.

	aggression	play	foraging	reunion
aggression	0.51	$7.48e^{-11}$	0.02	$1.45e^{-13}$
play	$< e^{-14}$	0.79	$1.46e^{-7}$	0.01
foraging	$< e^{-14}$	$< e^{-14}$	0.98	$1.63e^{-7}$
reunion	$3.00e^{-9}$	$2.58e^{-4}$	$< e^{-14}$	0.84

.

Table 8: The Likert scale results for every condition.

	Intra-Cluster	Inter-Cluster
Small Approx	4	4
Small Exact	3	3
Large Approx	6	3
Large Exact	5	6

CHAPTER VII

DISCUSSION

In the previous chapter, I presented several experiments evaluating the signal imager system. The system evaluates the exact and approximate algorithms. The annotation experiment showed that it is possible to annotate dolphin communication sequences with high accuracy using several algorithms and multiple statistics. In a more detailed analysis, I showed that the algorithm returns annotations that match visual behavior. In a breakdown of the percentage of true positives, I showed that the algorithms are able to annotate correctly when there is enough data. Furthermore, the support vector machine and the random forest show high precision. These results are the first indicator that the patterns found in the audio data are meaningful. The interesting result is that the system finds patterns in the audio data that are predictive of visual observations. For example, the system predicts multiple visual identifiers correctly. From the domain experts, we know that dolphins use signature whistles, which can be thought of as names, to call for each other. The system might use audio patterns that capture signature whistles to annotate a dolphin's identifier. Another possibility is that the system might use audio patterns associated with aggression to annotate visual behavior such as head-to-head.

Later experiments reveal that the best-performing statistics are the bag-of-words approach combined with the regular expressions. In a discussion with the domain expert, we agreed that the regular expressions can capture the sequential structure of dolphin communication, even under noisy conditions. In particular, the rules can deal with the insertion or deletion of patterns into a sequence, while n-grams cannot.

There are several possibilities why the rules perform well. First, the rules might capture actual sequential structure in the rules. Another possibility is that the rules can capture overlapping dolphin communication. If two dolphin groups communicate separately, both communications will show in the audio files, and the rules can separate the communications with the logic or between patterns and the gaps. The annotation experiment reveals that the exact algorithm performs better than the approximate algorithm. However, the results are only weakly significant.

Furthermore, the experiments show that the system is able to perform comparative statistics between the audible communications in several contexts. I use a ground truth dataset to evaluate if the system is able to find significant differences between different contexts. As shown in the results, comparing the contexts to each other, results in highly significant differences. Furthermore, comparing a context to itself results in highly non-significant differences. In other words, the system is able to determine the differences in the audible communication in different contexts. The results of the experiments indicate that the discovered patterns and their distributions are a good model for dolphin communication. There is another interesting discovery in these experiments: adding more data to the pattern discovery process results in better statistics. My intuition is that a large dataset is desirable for pattern discovery. The patterns have more data support since each pattern occurs more often in the dataset. The parameters of hidden Markov models become more stable when estimated from more data. A more stable estimate of the hidden Markov models' parameters also results in a better decoding of sequences and, in turn, a better estimate of the statistics. As seen in the experiment, adding data to the system removes the single error from the statistics. Furthermore, the differences between each context become more significant with more data, and the differences between the contexts to themselves become less significant. In other words, the results become more stable and agree more with the domain expert's opinion.

In the qualitative experiments, the domain expert seems to agree with the observations from the numeric experiments. First, the Likert scale indicates that the domain expert rates the patterns discovered in the big datasets higher. From her comments, it seems that she also rates the importance higher since she notes the difference between the datasets going towards communication units. Interestingly, the domain expert rates the exact algorithm for the slow dataset lower than the approximate counterpart. One explanation is that the exact algorithm needs more data to establish a good initialization for the hidden Markov models. Since the exact algorithm is based on clustering, fewer data points might lead to unstable clusters. Since the approximate algorithm is based on hashing coarsely quantized sequences, having fewer data points will not matter as much. Furthermore, during the qualitative exploration session the domain expert noticed some previous undiscovered patterns. For example, in the experiment she noticed two equal patterns next to each other and noted that she did not notice something like it before. The two patterns happened during play behavior with sargassum. The patterns for the small data set and the large data set are shown in Figure 36.

The domain expert notes that it seems like the patterns in the small dataset group burst pulse sounds, while the patterns in the larger dataset model the frequency modulation, resulting in a more detailed view of the data. Again, the expert's observations seem to confirm that a larger set of data stabilizes the pattern discovery. Another interesting comment from the domain expert is that it is easier for her to analyze a pattern set that is too inclusive than a pattern set in which the examples of each pattern are fully distinct. In her opinion, she prefers clean or crisp patterns. Her solution to multiple similar patterns is that she labels these patterns the same in her analysis. Splitting patterns is harder for her.

In the last experiments, I showed some common errors during the low-level processing of the data. It seems that some whistles are not traceable, especially when the

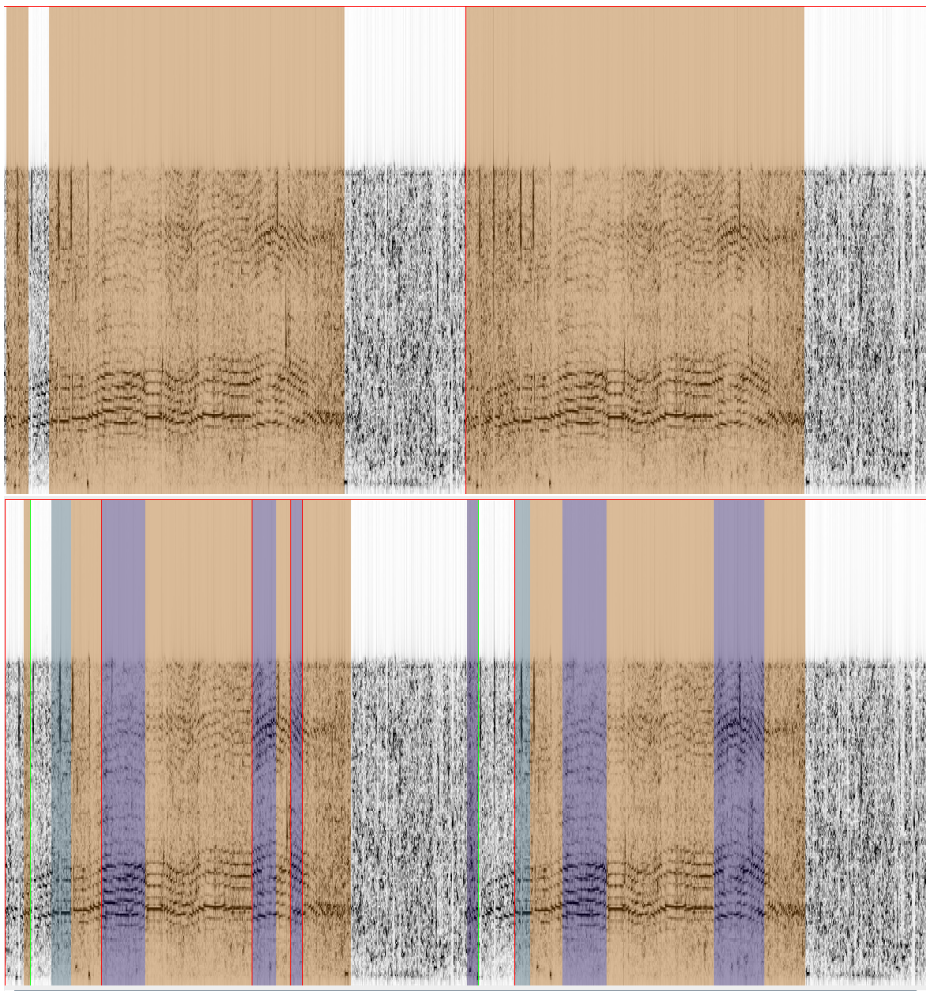


Figure 36: The patterns found in an audio file recorded during play behavior. Top: The patterns found using the small dataset. Bottom: The patterns found using the large dataset.

target whistle has parts indistinguishable from a noisy background. In another case, whistles were not traceable if there was a very strong noise source in the recording pulling the whistle trace towards it. The first problem can be solved by not adding the whistle to the categorized dataset or by splitting the whistle into pieces that are easily traceable. In the second case, pre-filtering the signal using a high-pass filter can lead to a better trace. Often the strong noise sources include boat noise, which is lower in frequency.

In an experiment with the signal detector, I found high accuracies when classifying

frames in the new feature space into signal or noise. However, when evaluating the segmentation results, I observed errors such as framing, false positives and false negatives. A framing error occurs when a pattern is split in the middle. Missing a pattern is called a false negative and inserting a pattern is called a false positive. However, the evaluation of the system's high level functions such as discovery, annotation and statistical testing reveals stability with respect to these errors. My hypothesis is that the hidden Markov models compensate for the errors in the segmentation. A mixture of hidden Markov models is decoding the sequences in total instead of the frame-by-frame signal detection. Framed patterns might match a complete pattern in the hidden Markov model, and false positives as well as false negatives will be reassigned to new hidden Markov model components.

During the experiments, the exact algorithm is slower than the approximate algorithm. On a standard laptop with an Intel Core i5 processor with $2.8GHz$ clock speed, a three MB L3 cache and four GB of RAM, the exact algorithm needs several hours to complete while the approximate algorithm needs about one hour. However, during the evaluation with the domain expert, both algorithms increased the discovery speed significantly compared to the previously manual analysis. Furthermore, the algorithm can discover patterns from data distributed over several years. Such an analysis is very hard without the algorithm. As described in the prairie dog example in Chapter I, such an analysis can take several decades. With the statistical testing capabilities, the signal imager provides numeric evidence that can be used in future biology scientific publications. The pattern discovery algorithm can also provide a less biased estimate of the patterns in dolphin communication than the visual comparison and manual measurements in the spectrogram.

The domain expert's comments during the signal imager session for the qualitative experiment, as well as the testing and annotation results, suggest that the

program is useful to the marine biology community. The experiments show the hypothesis testing and annotation capabilities using the results of the pattern discovery, the feature learning and communication models. Inspecting the patterns found in the datasets showed the capability to generate new research ideas regarding the patterns and helped the expert to gain insight into some interesting artifacts of dolphin communication.

As shown in the experiments, the signal imager performs very well in situations where the patterns of communication and the structure of the communication are unknown. From a set of audio files with contextual behavior annotations, the signal imager helps to uncover some of the communication patterns and helps biologists to interactively explore their data to gain insight into the unknown communication. However, one might ask how useful the signal imager is in domains with known structure and patterns such as human speech. In human speech, all the patterns are known in the form of words. Furthermore, the grammatical structure of human communication is known. Even if a formal grammar for human language is not found, the grammatical rules can be explained by humans. Furthermore, for specific domains such as telephone numbers or meeting scheduling [30], the definition of a formal grammar is not just possible but common. In other words, in domains with extensive domain knowledge and a high degree of confidence in such knowledge, the signal imager will perform worse than a fine-tuned system.

Take a speech recognizer for a specific domain as an example. All the words will be known as well as the syntactic structure. Furthermore, the words used and the structure do not have to model natural human speech but can be artificially engineered to increase recognition performance. Furthermore, collecting data for that recognizer can be performed in a highly directed manner since all the words needed for it are known. In fact, the resulting speech recognizer will be very similar to the mixture of hidden Markov models that I use to describe each pattern. However,

there is no need for the discovery algorithm since we already know which words to expect. Furthermore, the knowledge of human speech goes so far that even the features are engineered to match the human vocal tract (linear predictive coding) and auditory system (Mel-frequency cepstral coefficients). In conclusion, the signal imager performs best when the communication structure is unknown and a non-verbal analysis is the goal. Furthermore, the system provides a faster and richer analysis than possible with the state-of-the-art tools such as Cornell Lab of Ornithology's Raven.

CHAPTER VIII

FUTURE WORK

During the discussions with the domain expert, I identified several extensions to the signal imager as well as extensions to the experiments. On the biology side, one interesting possibility to extend this work is to repeat these experiments with different animal species. Other-well studied research fields that could benefit from the signal imager include zebra finch birds, prairie dogs and whales. The dolphin researchers themselves want to use the signal imager to analyze communication in several unexplored contexts such as inter-species aggression or communication across different ages. For example, the researchers plan to evaluate communication patterns during aggressive behavior between bottlenose dolphins and Atlantic spotted dolphins. Furthermore, the researchers want to correlate the audible communication patterns with body postures. One interesting extension to the signal imager is to include the visual dolphin features into the analysis. In that way, the correlation analysis can be performed in the same tool. Another interesting functionality is to use the signal imager to annotate communication and context identification in environments with restricted visibility. The system could help to study dolphin behavior even when visual inspection is not possible.

In order to use the results from the signal imager in the field the pattern model, statistics, and annotation classifiers as well as the contextual analyzers have to be exported and integrated into portable equipment. One promising target platform to deploy the model to is the CHAT platform. The Cetacean Hearing and Telemetry (CHAT) platform is an underwater wearable computer that supports dolphin communication field experiments [23].

Another system could use the annotation algorithms to find specific annotations in large collections of audio files. Searching through continuous audio streams such as a large database or an online recognizer requires a low false positive rate and high precision. Encouragingly, from the experiments I learned that the support vector machine and the random forest show high precision which indicates a low false positive rate. These search tools might enrich the biologist's field work and speed the search for specific audio files showing an annotation.

Another idea from the behavior researcher is to explore patterns at multiple levels. I can image changing the signal imager to support multiple resolutions of patterns. For example, saving the results along the hierarchical clustering dendogram could allow the building of several models, one at each dendogram level. Researchers can then interactively change the granularity of the system and inspect the patterns at multiple resolutions. Another experiment I propose is to compare the results of a discriminative approach to the unsupervised approach proposed in this thesis. A discriminative approach might be harder to analyze, and the generative approach in this thesis offers easy visual inspection of the patterns by domain experts. However, it would be interesting to compare the annotation performance using a purely discriminative approach to the annotation performance by my classifiers.

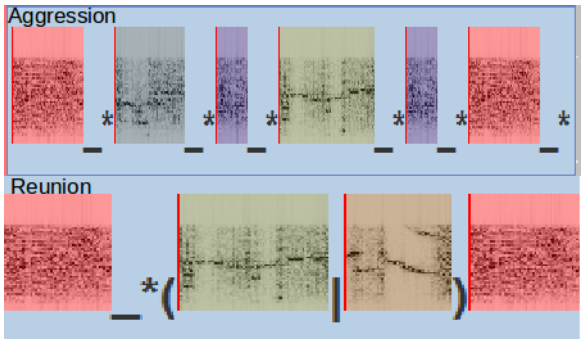


Figure 37: Two rules that match often. One matches often during aggression (Top); the other matches often during mother-calf reunion(Bottom).

Another interesting experiment for future work is to find out why certain rules

help to distinguish between different contexts. For example, aggressive behavior is often characterized by a certain burst pulse called a synchronized squawk. Often these sounds appear as short packages with a fairly regular rhythm. During mother-calf reunions, the prominent sounds are signature-whistles. In Figure 37 I show two example rules, one for aggression and one for reunions. The top one looks regular and is composed mostly of burst pulse patterns, while the reunion rule shows more whistle pieces. One experiment I propose is to search for meaning in the found rules together with a domain expert.

CHAPTER IX

CONCLUSION

This dissertation presented the signal imager. The signal imager is a program that can automatically detect patterns in audible dolphin communication. Furthermore, the program can use statistics including bag-of-words, n-grams and regular expressions to perform comparative statistics between different behavior contexts. Furthermore, the program can automatically annotate dolphin communication sequences. The signal imager uses feature learning to construct a novel feature space in which dolphin communication becomes comparable under frequency shift transformations. Furthermore, the system uses mixtures of hidden Markov models to discover time warped patterns. In a series of experiments I showed that the discovered patterns enable statistical testing and annotation. Furthermore, I presented results of a qualitative analysis with a domain expert that indicates that the discovered patterns are also meaningful to biologists. The results indicate that the algorithm pipeline produces patterns and statistics that provide insight into dolphin communication, can be used for retrospective analysis and can be used for statistical testing and scientific hypothesis generation.

APPENDIX A

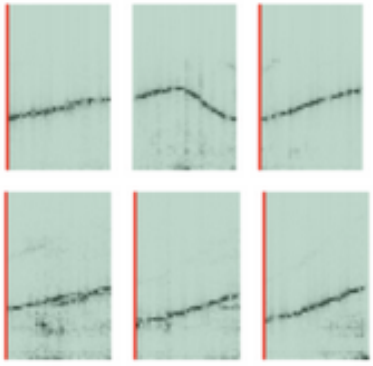
QUESTIONNAIRE FOR PATTERN EVALUATION

Evaluation of Different Algorithms

Cluster Tightness

Here we evaluate the perceived tightness of clusters. A cluster is tight if the examples in it are all similar to each other.

Example of Tight and Lose Clustering



Loose Clustering

Tight Clustering

Using method A results in a tight clustering

1 2 3 4 5 6 7

Strongly Diagree ○ ○ ○ ○ ○ ○ ○ Strongly Agree

Using method B results in a tight clustering

1 2 3 4 5 6 7

Strongly Disagree ○ ○ ○ ○ ○ ○ ○ Strongly Agree

Using method C results in a tight clustering

1 2 3 4 5 6 7

Strongly Disagree ○ ○ ○ ○ ○ ○ ○ Strongly Agree

Using method D results in a tight clustering

1 2 3 4 5 6 7

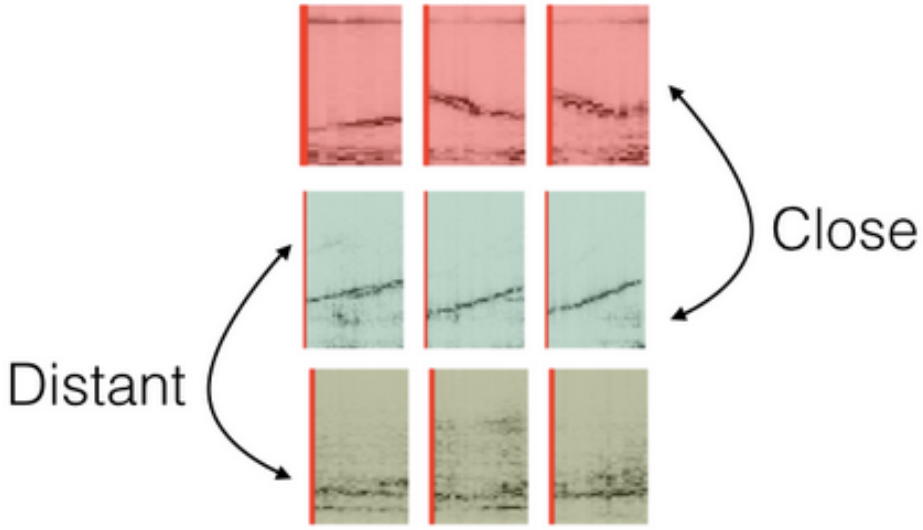
Strongly Disagree ○ ○ ○ ○ ○ ○ ○ Strongly Agree

Evaluation of Different Algorithms

Inter Cluster Distance

Here we evaluate the perceived distance between clusters. Two clusters are distant if the examples of one cluster are different from the examples in the other.

Distance of Clusters



Using method A results in a high inter cluster distance

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Using method B results in a high inter cluster distance

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Using method C results in a high inter cluster distance

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Using method D results in a high inter cluster distance

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

REFERENCES

[1] Adi, K., Sonstrom, K., Scheifele, P., and Johnson, M., "Unsupervised validity measures for vocalization clustering," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4377–4380, 2008.

[2] Baggenstoss, P. M. and Kurth, F., "Comparing shift-autocorrelation with cepstrum for detection of burst pulses in impulsive noise," *The Journal of the Acoustical Society of America*, vol. 136, no. 4, pp. 1574–1582, 2014.

[3] Bettadapura, V., Schindler, G., Ploetz, T., and Essa, I., "Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2619–2626, 2013.

[4] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.

[5] Breiman, L., "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] Buhler, J. and Tompa, M., "Finding motifs using random projections," in *Proceedings of the Fifth Annual International Conference on Computational Biology*, RECOMB '01, (New York, NY, USA), pp. 69–76, ACM, 2001.

[7] Castro, N. and Azevedo, P. J., "Multiresolution motif discovery in time series," in *SDM*, pp. 665–676, SIAM, 2010.

[8] Chiu, B., Keogh, E., and Lonardi, S., "Probabilistic discovery of time series motifs," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, (New York, NY, USA), pp. 493–498, ACM, 2003.

[9] Coates, A., Ng, A. Y., and Lee, H., "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

[10] Cornell, "Lab of ornithology's raven: Interactive sound analysis software," *Bioacoustics Research Program*, 2014.

[11] Deecke, V. B., Ford, J. K. B., and Spong, P., "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*, vol. 105, no. 4, pp. 2499–2507, 1999.

[12] Dennis, J., Tran, H., and Chng, E., "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[13] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G., *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[14] Esfahanian, M., Zhuang, H., and Erdol, N., "On contour-based classification of dolphin whistles by type," *Applied Acoustics*, vol. 76, no. 0, pp. 274–279, 2014.

[15] Halkias, X. and Ellis, D., "Call detection and extraction using Bayesian inference," *Applied Acoustics*, vol. 67, no. 11, pp. 1164–1174, 2006.

[16] Herzing, D. L., "Vocalizations and associated underwater behavior of free-ranging Atlantic spotted dolphins, Stenella frontalis and bottlenose dolphins, Tursiops truncatus," *Aquatic Mammals*, vol. 22.2, no. 2, pp. 61–79, 1996.

[17] Herzing, D. L., "Clicks, whistles and pulses: Passive and active signal use in dolphin communication," *Acta Astronautica*, vol. 105, no. 2, pp. 534–537, 2014.

[18] Hinton, G. E., "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade (2nd ed.)* (Montavon, G., Orr, G. B., and Müller, K.-R., eds.), vol. 7700 of *Lecture Notes in Computer Science*, pp. 599–619, Springer, 2012.

[19] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S., "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

[20] Kershenbaum, A., Sayigh, L. S., and Janik, V. M., "The encoding of individual identity in dolphin signature whistles: How much information is needed?," *PLoS ONE*, vol. 8, no. 10, 2013.

[21] King, S. L., Sayigh, L. S., Wells, R. S., Fellner, W., and Janik, V. M., "Vocal copying of individually distinctive signature whistles in bottlenose dolphins," *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1757, 2013.

[22] Kohlsdorf, D., Mason, C., Herzing, D., and Starner, T., "Probabilistic extraction and discovery of fundamental units in dolphin whistles," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 8242–8246, 2014.

[23] Kohlsdorf, D., Gilliland, S., Presti, P., Starner, T., and Herzing, D., "An underwater wearable computer for two way human-dolphin communication experimentation," in *Proceedings of the 2013 International Symposium on Wearable Computers*, pp. 147–148, ACM, 2013.

[24] Lampert, T. A. and O'Keefe, S. E., "An active contour algorithm for spectrogram track detection," *Pattern Recognition Letters*, vol. 31, no. 10, pp. 1201–1206, 2010.

[25] Lampert, T. A. and O'Keefe, S. E., "A survey of spectrogram track detection algorithms," *Applied Acoustics*, vol. 71, no. 2, pp. 87 – 100, 2010.

[26] Lawrence, E., Altschul, F., Boguski, S., Liu, S., F., N., and Wootton, C., "Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment," *Science*, vol. 262, no. 5131, pp. 208–214, 1993.

[27] Lee, H., Battle, A., Raina, R., and Ng, A. Y., "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, pp. 801–808, 2007.

[28] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 609–616, 2009.

[29] Lin, J., Keogh, E., Wei, L., and Lonardi, S., "Experiencing SAX: A novel symbolic representation of time series," *Data Mining Knowledge Discovery*, vol. 15, pp. 107–144, Oct. 2007.

[30] Lyons, K., Skees, C., Starner, T., Snoeck, C. M., Wong, B., and Ashbrook, D., "Augmenting conversations using dual-purpose speech," in *Proceedings of ACM User Interface Software and Technology (UIST) 2004*, (Santa Fe, NM), pp. 243–246, 2004.

[31] Miles, H., "Underwater analysis of the behavioral development of free-ranging Atlantic spotted dolphin Stenella frontalis) calves (birth to 4 years of age)," *Aquatic Mammals*, vol. 29, no. 3, pp. 363–377, 2003.

[32] Minnen, D., Isbell, C., Essa, I., and Starner, T., "Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery," in *Proceedings of 2007 Seventh IEEE International Conference on Data Mining (ICDM)*, pp. 601–606, 2007.

[33] Minnen, D., Isbell, C., Essa, I., and Starner, T., "Discovering multivariate motifs using subsequence density estimation and greedy mixture learning," in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pp. 615–620, 2007.

[34] Minnen, D., Isbell, C. L., Essa, I., and Starner, T., "Discovering multivariate motifs using subsequence density estimation," in *AAAI Conference on Artificial Intelligence*, pp. 615–620, 2007.

[35] Mitarai, S., Hirao, M., Matsumoto, T., Shinohara, A., Takeda, M., and Arikawa, S., "Compressed pattern matching for SEQUITUR," in *Data Compression Conference*, pp. 469–478, IEEE Computer Society, 2001.

[36] Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., and Westover, B., "Exact discovery of time series motifs," in *SIAM International Conference on Data Mining (SDM09)*, pp. 473–484, American Statistical Association (ASA), 2009.

[37] Park, A. and Glass, J. R., "A novel DTW-based distance measure for speaker segmentation," in *Spoken Language Technology Workshop (SLT)*, pp. 22–25, IEEE, 2006.

[38] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, pp. 257–286, 1989.

[39] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y., "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 759–766, 2007.

[40] Russell, S. J. and Norvig, P., *Artificial intelligence: a modern approach (3rd edition)*. Prentice Hall, 2009.

[41] Saria, S., Duchi, A., and Koller, D., "Discovering deformable motifs in continuous time series data.," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI11)*, pp. 1465–1471, 2011.

[42] Shapiro, A. D. and Wang, C., "A versatile pitch tracking algorithm: From human speech to killer whale vocalizations," *The Journal of the Acoustical Society of America*, vol. 126, no. 1, pp. 451–459, 2009.

[43] Shieh, J. and Keogh, E., "iSAX: Indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (New York, NY, USA), pp. 623–631, ACM, 2008.

[44] Shokoohi-Yekta, M., Wang, J., and E., K., "On the non-trivial generalization of dynamic time warping to the multi-dimensional case," in *SIAM International Conference on Data Mining (SDM15)*, 2015.

[45] Sipser, M., *Introduction to the Theory of Computation (1st ed.)*. International Thomson Publishing, 1996.

[46] Slobodchikoff, C. and Placer, J., "Acoustic structures in the alarm calls of Gunnison's prairie dogs," *Journal of the Acoustical Society of America*, vol. 119, no. 5(1), pp. 3153–3160, 2006.

[47] Smyth, P., "Clustering sequences with hidden Markov models," in *Advances in Neural Information Processing Systems*, pp. 648–654, MIT Press, 1997.

[48] Stolcke, A. and Omohundro, S., "Hidden Markov model induction by Bayesian model merging," *Advances in neural information processing systems*, pp. 11–18, 1993.

[49] van Zaanen, M., "ABL: Alignment-based learning," in *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 961–967, 2000.

[50] Verma, Y. and Jawahar, C., "Image annotation using metric learning in semantic neighbourhoods," in *ECCV Proceedings 2012*, pp. 836–849, 2012.

[51] Wang, A., "An industrial-strength audio search algorithm," in *Proceedings of the 4th International Conference on Music Information Retrieval*, pp. 7–13, 2003.

[52] Zakaria, J., Rotschafer, S., Mueen, A., Razak, K., and Keogh, E., "Mining massive archives of mice sounds with symbolized representations," in *Proceedings of the International Conference on Data Mining*, pp. 588–599, 2012.

[53] Zhou, F., De la Torre Frade, F., and Hodgins , J. K., "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, pp. 582–596, March 2013.