Methods for Discovering Models of Behavior:
A Case Study with Wild Atlantic Spotted Dolphins

Daniel Kohlsdorf[1], Denise Herzing[2], and Thad Starner[1*]

[1]Georgia Institute of Technology, Atlanta, GA
[2]Wild Dolphin Project and Florida Atlantic University, Boca Raton, FL

*Corresponding author (Email: thad@gatech.edu)

Abstract - We present an interactive data mining method to support marine mammalogists investigating communicative behaviors with wild Atlantic spotted dolphins, *Stenella frontalis*. Most current tools require scientists to inspect behavior data manually, form hypotheses, create a detector for patterns of interest, search their database of behavior, and measure the results to determine if their hypothesis was correct. In contrast, our technique discovers patterns in the data automatically, allowing researchers to associate these patterns with behaviors of interest. By visualizing where in time these patterns occur and in what orders, the scientist can quickly form hypotheses and then use statistical tests on a reserved test set of data to determine whether those hypotheses have support. Our hope is that these methods will speed behavioral research significantly.

Keywords – Dolphin communication modeling, Pattern discovery, Hidden Markov models, Machine learning

---

Behavioral researchers often collect large amounts of data that can be difficult and time-consuming to analyze. Often analysis is done visually, playing through recordings repeatedly to first determine which behaviors should be annotated and then marking the beginnings and ends of those behaviors. Our conversations with dolphin communication researchers suggest that every hour of animal behavior requires 10 hrs of manual analysis. We found similar estimates of time spent when working with behavior researchers examining autism, activities of daily living, and industrial behavior. This 10 to 1 ratio of analysis time to data collection time occurred in our own work developing computer recognition systems for handwriting, American Sign Language, and gestural interfaces.

For behavioral researchers, such slow analysis leads to delays in publication. For example, in studying the communication patterns of Gunnison's prairie dogs (*Cynomys gunnisoni*), researchers suspected that the animals code information about the type of threat in their alarm calls. However, the research process to establish this fact took decades from initial data collection to publication (Slobodchikoff & Placer, 2006). This paper addresses new techniques, taken from other pattern recognition programs, that might be helpful in the analysis of large, undiscovered datasets of marine mammals.

Background Work
Solving the Problems

Speeding analysis with current tools. One way of speeding analysis is to increase the number of researchers examining the data. However, this approach is costly and may lead to questions of consistency. Considerable training may be required so that the analysis can be performed in a repeatable way. For situations where labeling requires a value judgment, inter-rater reliability becomes important, and multiple researchers may examine the same data independently and then compare their annotations to provide a "ground truth" labeling. This process requires even more time. For some behavioral researchers, large subsets of the data may remain unexplored for years or even decades and, even when analysis is performed, it is often done looking at a very narrow range of behaviors.

A common aid for the manual analysis of animal acoustic communication are interactive computer visualization programs such as Cornell Lab of Ornithology's Raven (http://www.birds.cornell.edu), Noldus Information Technology's Observer (http://www.noldus.com). XBAT (xbat.org), Sound Analysis Pro (soundanalysispro.com), Praat (www.praat.org), and Adobe Audition (www.adobe.com/audition). Noldus Observer enables behavioral researchers to code behavior with annotations in video and audio. Recently, Observer added full-spectrum audio analysis tools through an additional product called Ultravox. Our discussions with, and observations of, animal vocalization researchers show that they often use more general purpose tools like Observer and Audition to provide an overview of their results, whereas tools like Raven, XBAT, and Sound Analysis Pro are more for inspecting the spectrogram, taking measurements, and creating and investigating hypotheses. Of particular interest, Raven, XBAT, and Sound Analysis Pro allow the user to hand specify detectors or train them from examples to attempt to identify regions of interest in the spectrogram and speed analysis. Here we focus on Raven as an example of these tools because it is commonly used.

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. Unfortunately, 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. Raven also includes techniques such as correlation to compare signals between spectrograms. However, correlation does not compensate for transformations such as frequency shifts and time warping effects. In practice, these tools require significant manual effort from the user; the animal communication researchers we have observed use Raven primarily as a tool to compare vocalizations visually and to assist with manual measurements on the spectrogram. Although Raven is designed to be extensible, it is not open source, which makes contributing to the project difficult.

Speeding analysis with pattern recognition algorithms. In addition to the tools described above, animal vocalization and behavior researchers often adapt pattern recognition techniques to help identify and classify classes of behavior. Just as with the tools mentioned in the previous section, most pattern recognition and machine learning approaches in the literature require that the researcher specify what type of event they wish to locate in the data and provide training examples, or measure specific features that can be used to segment the desired behavior.

For underwater acoustics, Lampert and O'Keefe's (2010a) survey identifies three main algorithm categories for analyzing the spectrogram: 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. Alternatively, Kershenbaum, Sayigh, and Janik (2013) measure the similarity between whistles using the dynamic time warping distance. This task can be performed automatically on a variety of dolphin signals, as shown by recent efforts of Baggenstoss

and Kurth (2014) who compare methods for detecting burst pulses. Kohlsdorf, Mason, Herzing, and Starner (2014) extract a dolphin whistle using a probabilistic pitch tracker. Other approaches to whistle extraction include a frame-based Bayesian approach (Halkias & Ellis, 2006) and a Kalman filtering approach (Lampert & O'Keefe, 2010b). Shapiro and Wang (2009) apply pitch detection designed for human telephone speech to whale vocalizations. Dolphin signal clustering and classification often uses neural networks (Deecke, Ford, & Spong, 1999; Esfahanian, Zhuang, & Erdol, 2014) or clustering based on hidden Markov models (Adi, Sonstrom, Scheifele, & Johnson, 2008). Both of these neural network and HMM approaches filter the data first and use Mel-cepstral coefficients or other measurements from the spectrogram as features.

Speeding analysis with behavior discovery. Our goal is to improve researchers' efficiency and efficacy of categorizing clicks, whistles, and bursts in dolphin vocalizations (Herzing, 2000) significantly beyond current methods. To reach this goal, we are extending a general behavior discovery framework that we have used successfully in analyzing human behavior (see below).

For dolphin vocalizations we adopt an approach similar to the work done by Zakaria and colleagues on mining archives of mouse sound using symbolic representations (Zakaria, Rotschafer, Mueen, Razak, & Keogh, 2012). 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. Our system combines several methods from previous work, and Figure 1 provides an overview of the system we will describe in this paper. In brief, image-processing methods similar to convolutional neural networks are used as the basis for feature extraction (Coates, Lee, & Ng, 2011). We use feature learning on a dataset of several patches found around traced whistles and other dolphin signals. We then use hierarchical clustering to identify units. Similar to Zakaria, we proceed to process the units as discrete strings. However, instead of using an alignment score (Levenshtein distance) we use the alignment to extract a set of regular expressions, which can then be used to compare different vocalizations. In contrast to most dolphin communication research, our system is able to process a wide range of dolphin signals, not just whistles.

Behavioral Discovery in Other Fields

Towards behavior discovery. Behavior discovery is the unsupervised identification and modeling of actions embedded in a larger sensor stream. As discussed above, most approaches to recognizing human or animal activities learn models from hand-labeled sequences. Researchers decide a priori which components are important and either manually construct detectors or label training data for use with machine learning methods that create models appropriate for recognition. Behavior discovery inverts this problem and attempts to discover the important components through unsupervised analysis of the captured data.

Utilizing the discovery process has three main benefits. First, in situations where researchers believe they know the important constituent actions associated with a behavior, an unsupervised discovery system can either provide validation or suggest alternatives. Although unsupervised methods can be biased by initial conditions, consistent convergence upon a set of constituent actions over a range of initial conditions does provide support that the discovered actions are present and distinct in the data. When automatic analysis matches the intuition of the researcher, then the researcher can be more confident in using those constituent parts to detect and analyze behavior.

Secondly, when researchers do not know the low-level components of a behavior, a discovery system may help identify them. This benefit may be especially useful when a researcher has acquired large databases of behavior and does not have the ability to inspect the data by hand. A discovery system can help researchers inspect data, form hypotheses more quickly, and test those hypotheses.

Finally, researchers may not be able to identify or describe actions that are easily detected by a given system's sensors. This challenge is especially true when the phenomena being sensed are outside the normal human sensory range, as with ultrasonic sound or infrared light. Even with a spectrogram of

normal human speech, a researcher requires expert training to recognize words or even individual phonemes without the benefit of playing the corresponding audio. Fortunately, discovered actions are necessarily recognizable by the discovery system.

In summary, the goals of a behavior discovery system include validating a priori assumptions, generating new knowledge about a dataset, and interpreting data from sensors that might otherwise overwhelm the researcher. To make these issues more concrete, we describe previous work in applying discovery to human behavior where ground truth is mostly known or easy to determine. Then we describe how we have adapted our methods to address dolphin vocalizations where we do not necessarily know ground truth and report early results.
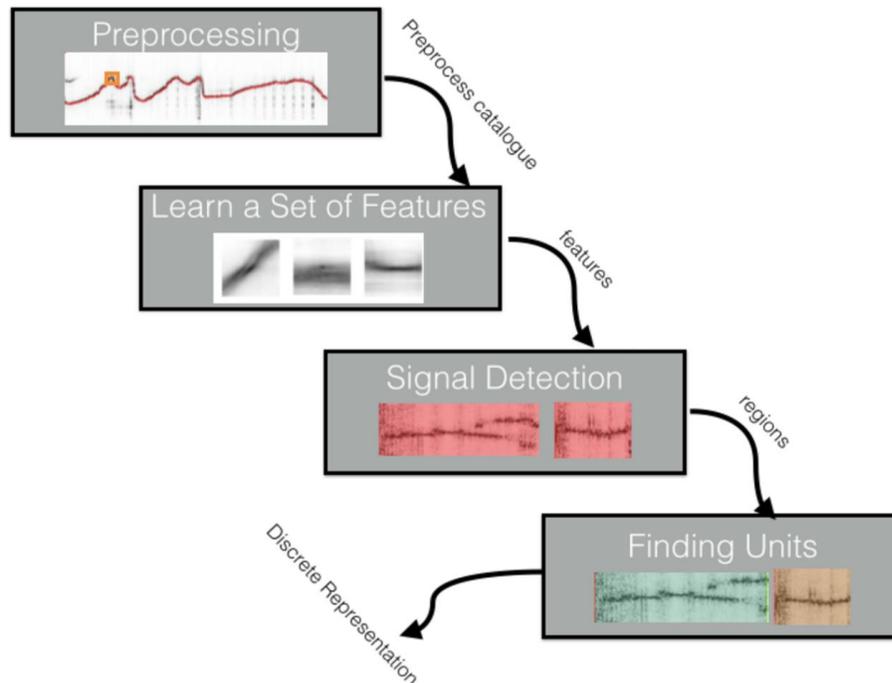
Figure 1. Our dolphin signal processing pipeline. Audio files containing only dolphin vocalizations are converted to spectrograms. The spectrogram around the dolphin vocalizations are divided into patches approximately 1 ms in time and 1 kHz in frequency. These patches are clustered, and the top 60 most common are used as features to segment dolphin vocalizations into regions. The features are also used as the basis for hierarchical clustering to identify units in these regions. These units provide a discrete representation of the dolphins' vocalizations, and we align these discrete representations to extract a set of regular expressions. We have used these regular expressions to predict the labels biologists give the dolphins' behaviors.

Discovery of locations and predictions of destinations from streams of continuous Global Positioning System (GPS) data. From a machine-learning standpoint, behavior discovery systems are difficult to create due to their reliance on unsupervised training; the classes of behavior are not known and no training examples are given. Fortunately, many behavioral actions are well separated in space or time or both, which means the classes are often distinct and can be discovered.

Ashbrook and Starner (2003) demonstrated an early example of behavior discovery where their algorithm could examine a user's GPS data to automatically find meaningful locations and predict the user's next destination. This unsupervised method works due to the spatial separation of locations in the user's daily routines and due to the visits to these locations being separated in time (i.e., the user can not be in two places at once and time was required to transition from one location to another). Beginning with streams of GPS latitude and longitude data (Figure 2a), the authors retained only places where the user stopped for more than 10 minutes (Figure 2b). Groups of these points were then clustered, and the clusters were defined as locations (Figure 2c). Finally, the participant's history of movement from one location to

another was mined to create a predictive model of how the user might move in the future (Figure 2d). The model's accuracy improved when both the current location and the previous location were used to predict the user's future location (i.e., a second order Markov model). An example of a practical use of such a model is a mobile phone application that predicts the user's next destination and alerts the user if she needs to leave early due to traffic. In the past few years, Google Now started providing a similar service on Android phones.

Interestingly, the Ashbrook and Starner method was used recursively to recover a hierarchy of meaningful locations at different scales automatically. Details are provided in the reference. At a high level, when locations are clustered into areas with a 20-mile radius, the system recovers cities. If the locations within a city are then clustered, the system returns campuses. Clustering locations within campuses returns buildings. In theory, continuing the process with the addition of an indoor localization system would return entrances or rooms. Recovering this hierarchy of scale shows that some behaviors are well enough separated in time and space that unsupervised algorithms are viable. Can similar techniques be created to discover more complex behaviors in continuous streams of data?

Discovery of specific exercises from wrist-mounted motion sensors. In the GPS case above, a discovery system found meaningful locations and then constructed a predictive model of behavior by examining the occurrence of sets of those locations through time. In such a system, the locations were discrete and relatively few in number. Discovering repeated patterns in the data could be performed by simply tracking all possibilities as they occurred. However, many behaviors are composed of actions that are continuous instead of discrete. Such actions may be performed slowly or quickly or change in speed and still fill the same function. Creating a system that can discover such actions, even when they have variations in their time of performance, and use them to uncover higher-level behaviors is difficult.

In his dissertation, Minnen (2008) presents several algorithms that can perform such a task and tests these algorithms on speech, sign language, optical character recognition, and several other domains. Minnen's algorithms, which are closely related to the ones we use for analysis of dolphin vocalizations below, are uninformed to the type of data they are receiving, and the number of units (spoken digits, signs, typed characters, etc.) that comprise the data.

The goal of the system is to discover these units and mark where they occur in the data set. One of the most illustrative data sets from the perspective of this discussion is his work on discovering the actions involved in an exercise routine involving six exercises with dumbbells. Motion sensors, equivalent to what is found in modern smartwatches, monitored the participant's arm motions during the exercise routine. Data are collected at 100 Hz, which is more than sufficient to capture exercises where each repetition requires around two seconds. Figure 3a shows example traces from the sensors. The top section shows the x, y, and z components of the accelerometer. Accelerometers are particularly useful for distinguishing exercises as the acceleration due to gravity helps recover the orientation of the sensor with respect to ground. The next set of three lines show the rotation about the x, y, and z axes as sensed by the gyroscope. Inspecting the data, there seem to be repetitions that are grouped in sections. This regularity becomes even clearer after Minnen's algorithm quantizes the data. Figure 3b shows the color-coded representation of this quantization process. Each line represents one of the 32 examples of exercise routines. The algorithm automatically discovers seven motifs that, in this case, correspond to the six exercises (Figure 4) and a rest state. Interestingly, the same algorithm discovered individual signs in phrase level sign language, spoken digits, and words in songs and political lectures.
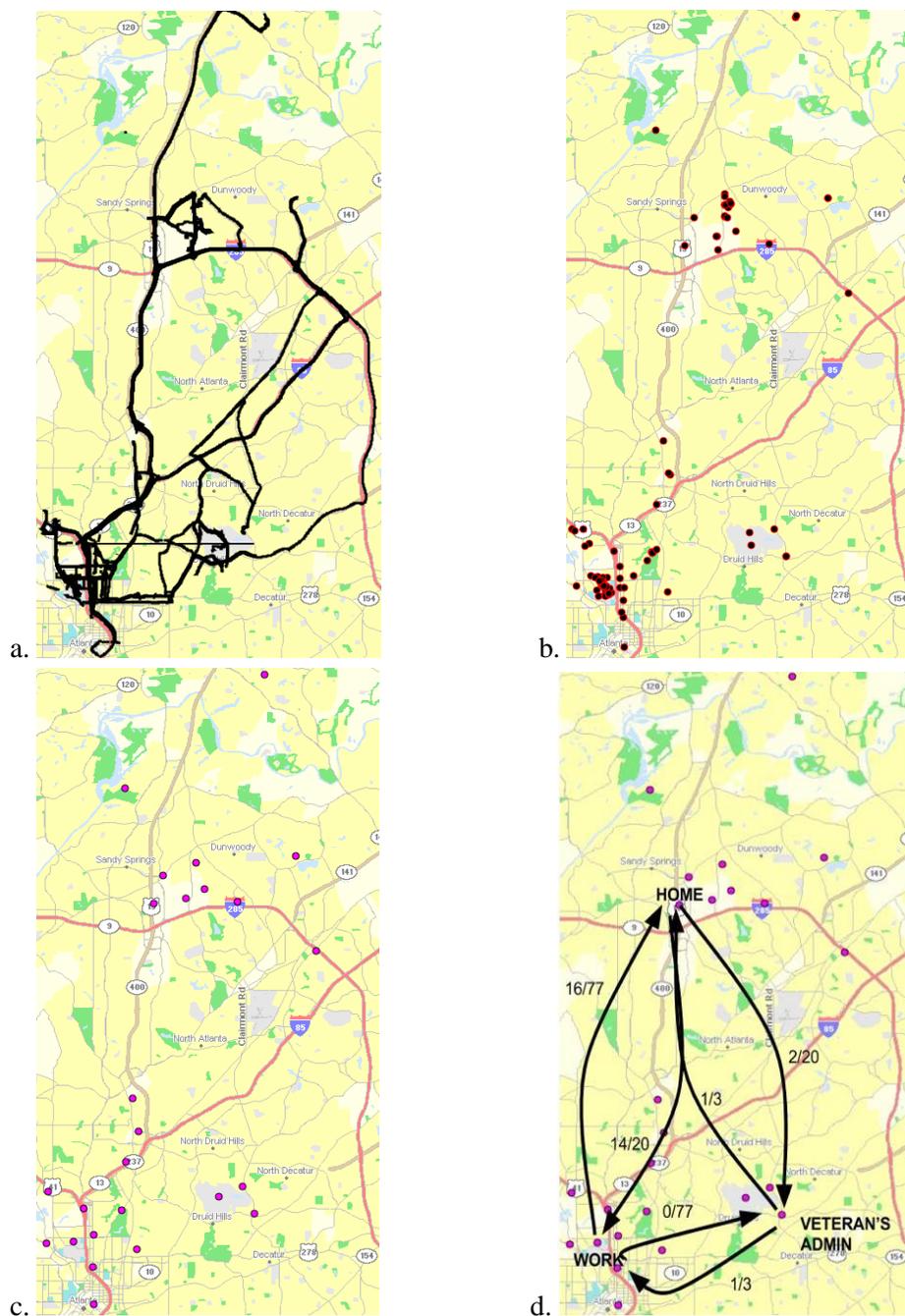
Figure 2. Discovering locations and routines of commuter movement. (a) an unfiltered time series of GPS data, (b) places where the user stops for more than 10 min, (c) clustering places into locations, and (d) creating a Markov model based on the user's past transitions between locations.

With the exercise data set, we knew a priori that there were six exercises, but suppose we had used the algorithm on the data without such knowledge. Further suppose we were told simply that the data came from motion sensors attached to primates. Given a rough estimate as to how long a motif might be (in this case, around one second), Minnen's system would return its seven motifs and mark where they occur in the data. The researcher could then examine each motif to determine if it is a behavior of interest and could collect overall statistics on the motifs. In this case, there were 144 examples of each of the six exercises with several examples repeated back to back (as makes sense with an exercise routine with

repetitions). This structure would be immediately obvious to the researcher. The researcher could make hypotheses based on some of the data (for example, motif 2 always follows motif 5) and check them quickly against a reserve part of the database or against other databases. For ongoing observations, the researcher could mark which motifs were interesting from the initial data set and have the system automatically label any examples it sees in future datasets or in live data.

If, on the other hand, behaviors of interest in the data set were already known, the researcher could determine which motif, or strings of motifs, typically corresponds to that behavior. Based on this expected mixture of motifs, creating a detector for labeling future examples of that behavior would be simple. In other words, a discovery system can help researchers quickly identify behaviors of interest and their constituent parts and create detectors that might be used to find specific instances of those behaviors in large observational datasets.

If we can produce a system that works on animal behavior even a fraction as well as has been shown on human behavior, it could speed analysis of large databases of animal behavior by an order of magnitude. For our first attempt of such a system, our team at Georgia Tech joined with the Wild Dolphin Project to determine if we could create a discovery system that would work on dolphin vocalizations.
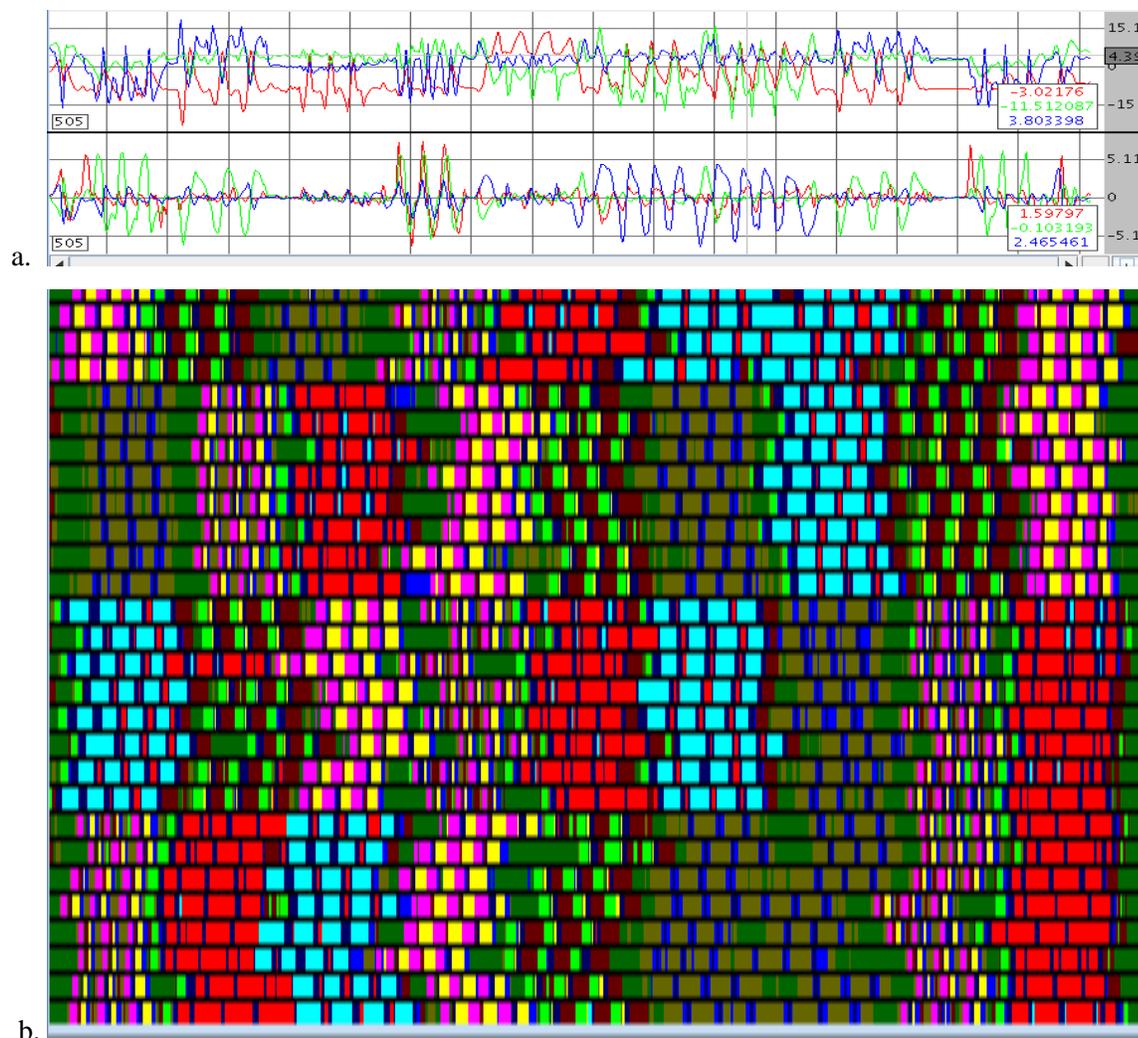
Figure 3. Discovering individual exercises in exercise sessions. (a) The user's wrist movements as measured by a 3-axis accelerometer and a 3-axis gyroscope are analyzed to discover repeated motifs. (b) These motifs are shown via unique colors in each of the 32 recorded exercise sessions. This simple visualization reveals the patterns of motifs that correspond to the six different exercises performed.
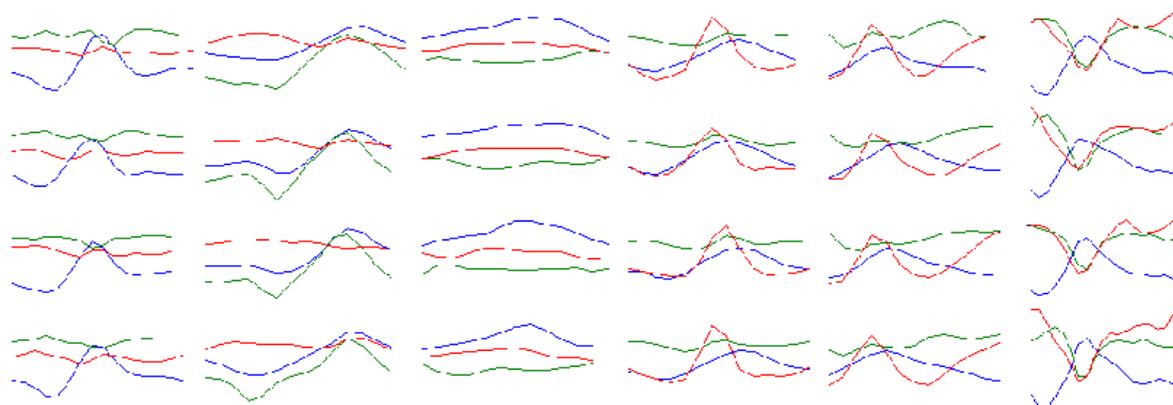
Figure 4. Examples of the six discovered exercises. Each column demonstrates four examples of the six exercises discovered from the 32 exercise sessions. Examination of the peaks and valleys reveals the temporal variability between the examples.

Using Discovery for Marine Mammals

Towards discovery in dolphin vocalizations. 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 communication 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. In the following, we present an interactive data mining system that supports marine mammalogists as they search multimedia databases more efficiently and inspect their data using statistical testing.

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 (Herzing, 1996) has collected over 30 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 as the collected data can provide insight into the social structure and communication patterns of a species.

The work of King, Sayigh, Wells, Fellner, and Janik (2013) shows the power of such vocalization databases. Over 200 hours of acoustic recordings of temporarily caught-and-released, wild bottlenose dolphins were initially annotated by one observer and segmented manually. The resulting analysis shows how dolphins can use whistles as names for themselves and others to maintain social bonds (King et al., 2013).

Search in audible dolphin communication. Our goal is 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.

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 rapid burst-pulse series can show as multiple parallel lines when the computer fails to differentiate individual clicks. Although these lines are technically an artifact of computer processing, the spacing between the lines represents the repetition rate of the clicks and can be a proxy for click rate.

As one can see in Figure 5, 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 of dolphin communication and its relation to other behavior.
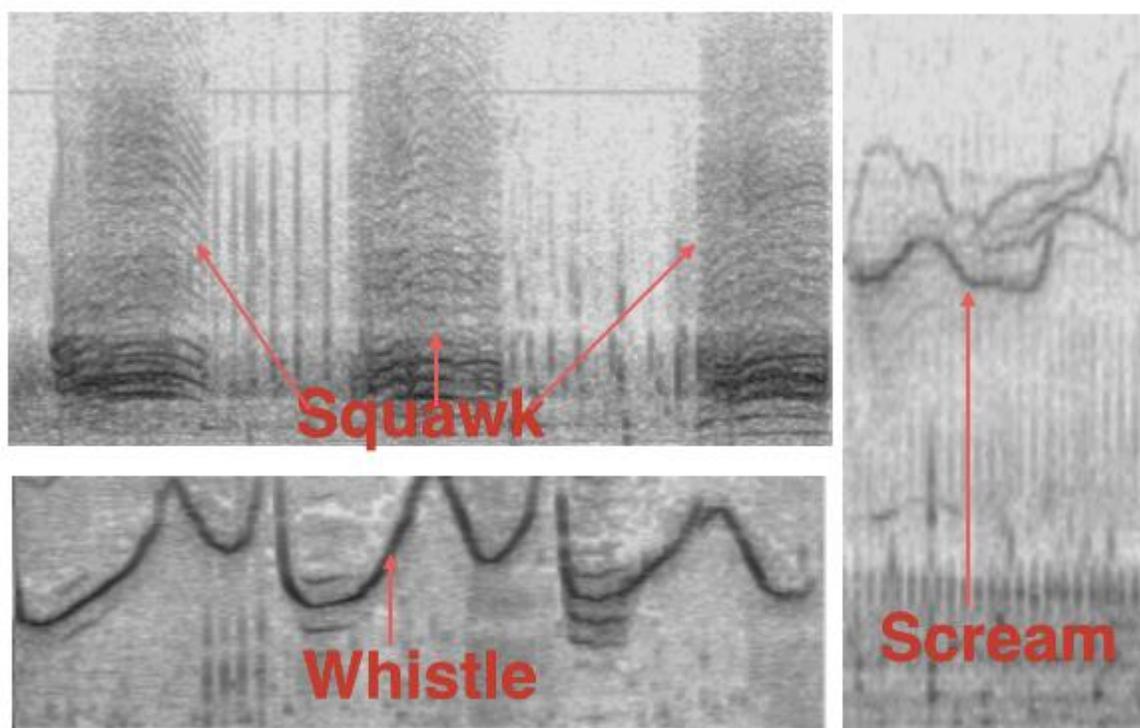
Figure 5. Different audible dolphin communication signals.

How the Algorithms Work

Signal imager: A brief algorithm overview. In this section, we describe the background for our system, called the Signal Imager, which we are designing for animal communication experts to more quickly analyze their databases using behavior discovery techniques. Although we have begun to use the Signal Imager on other databases, such as Zebrafinch vocalizations, our focus has been on dolphin communication. Here we discuss the design choices for our pattern discovery algorithm and show its ties to computer vision and speech recognition.

Our pattern discovery algorithm is invariant under two transformations that can be found in dolphin communication: frequency shifts and time warping. In other words, two dolphin signals should be treated as similar (given a particular 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 of 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 sounds maintain the same meaning- their spectral shape will remain uniquely distinguishable- allowing other humans the ability to recognize the words, or whistles, under these transformations. Furthermore, marine mammal research shows that dolphins perform similar transformations to their own whistles (see Harley, 2008; Richards, Woltz, & Herman, 1984; Sayigh, Esch, Wells, & Janik, 2007). An example of these transformations for a dolphin whistle is shown in Figure 6a.

Below we will discuss our choice of using convolutional features for pattern discovery and our approach to learning a set of feature extractors automatically. Afterwards, we describe how hidden Markov models can be used for automatic pattern discovery and how this method applies to dolphin communication.

a.

b.
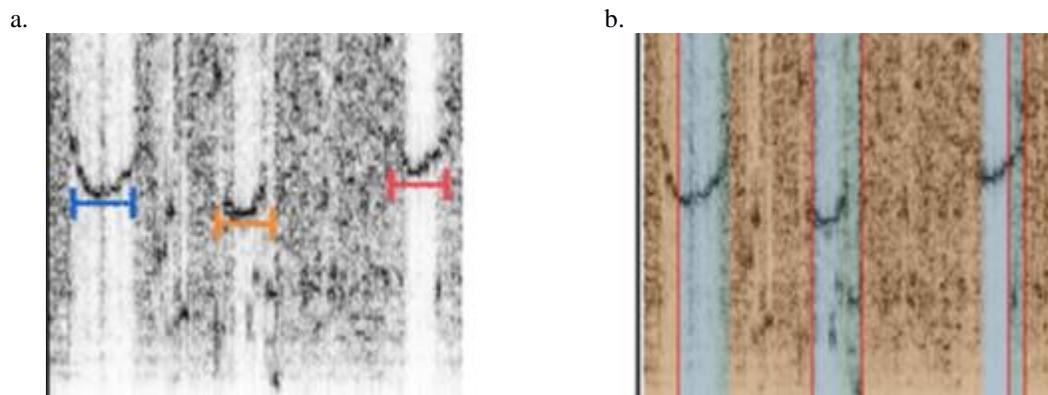
Figure 6. (a) An example where three patterns similar in shape vary in time and frequency. (b) A segmentation result from our algorithms.

Finding shift and warp invariant patterns. Researchers capture acoustic dolphin communication in digital field 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. Each point in the spectrogram represents the magnitude of a frequency at a specific time of the original audio wave. Because dolphin signals are inspected traditionally in a spectrogram, and modern speech recognition systems use this representation as a starting point for further feature extraction, we also search for patterns in the spectrogram. We define a signal pattern as a set of subsequences from several spectrograms that appear similar to each other given a distance function. Each pattern is an element of a global pattern codebook shared across all sequences.

In order to enable a frequency-invariant comparison of dolphin signals, we learn a set of feature extractors spanning a k-dimensional feature space. Using how well the features match the spectrogram, we can convert the spectrogram into a time series in the novel feature space with k dimensions and length T. We wish to learn features such that two dolphin signals that are similar in shape but are in different frequency bands appear close in the novel feature space under Euclidean distance. Furthermore, dolphin signals and other underwater noise sources should show distinct differences when compared in our feature space. Figure 7 illustrates the theory of our approach.

The algorithm for feature learning clusters small, local regions from the spectrogram using k-means (Coates et al., 2011) and transforms a novel spectrogram into the feature space using a soft k-means assignment. The soft k-means assignment computes a distance, from the spectrogram, of cluster regions and then converts these distances into an influence score for each cluster. The final feature space is

constructed by max pooling.
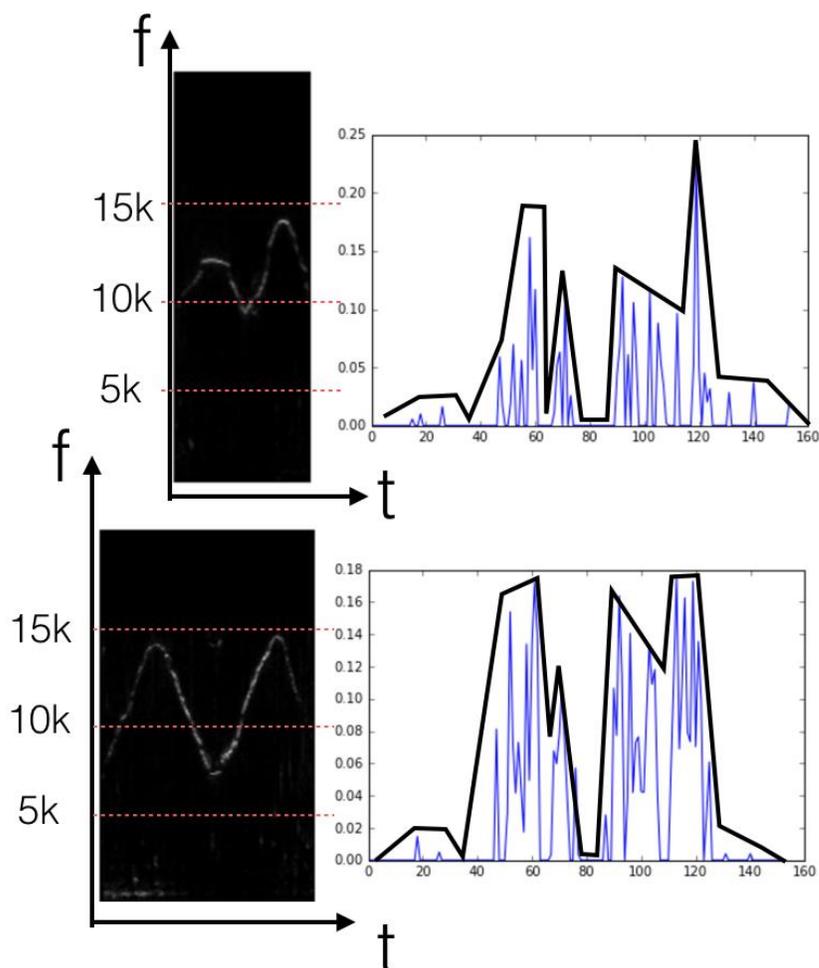
Figure 7. We wish to create a system that is insensitive to absolute frequencies when comparing vocalizations. We use convolutional features such that as long as some patch of the spectrogram at each time slice matches the feature, there will be a high response at that time. We then compare the response of all features over time using dynamic time warping to get a distance. Here, even though the bottom whistle shows much more variation in frequency, with the first peak almost reaching 15 kHz and the first trough reaching 7 kHz, the response to the feature through time is similar to the top whistle, resulting in a close distance score between the two.

The algorithm estimates the feature extractors from a dataset of dolphin signals with the following properties automatically:

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

2.   All audio files in the catalog are pure, meaning no audio file contains more than one signal type.

These properties will help us 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 includes only a dolphin signal. (This process can be assisted with an automatic filter that helps distinguish dolphin signals from other sounds, but, for the experiments described below, we depended upon signals that were segmented for previous unrelated experiments).

We transform each audio example in the catalog into its spectrogram representation. The main idea is to represent a feature extractor as a square region that represents a small portion of each signal. For example, a patch centered at one point on a dolphin whistle might capture a small part of an up sweep in frequency. A patch centered on a different location might capture a down sweep (Figure 8). Such a patch can be regarded as a local estimate in the spectrogram. In our experiments, a patch represents approximately half a millisecond in time and one kHz in frequency.
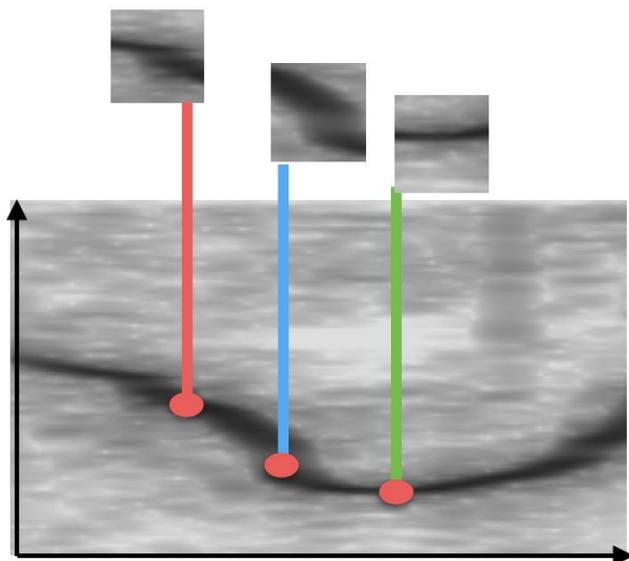
Figure 8. Three patches extracted around a dolphin signal. Two are around different down sweeps and one around a plateau.

Given the spectrograms extracted from such a catalog, we extract all patches that occur 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. We use unsupervised feature learning with k-means clustering (Coates, Lee, & Ng, 2011) to form a codebook of regions. The centers of 30 clusters learned from dolphin signal patches are shown in Figure 9. The resulting codebook represents our 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, we perform the following steps.
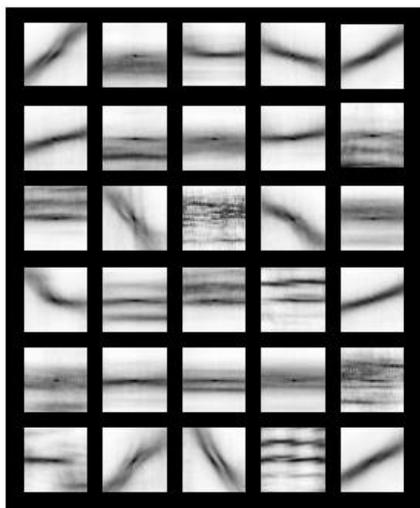
Figure 9. A set of 30 feature extractors learned using k-means. Ones with signal lines are from whistles whereas ones with

multiple lines are from bursts.

We place one of the clusters at a point in time and frequency in the spectrogram and compute the distance of the region to the spectrogram area it covers. We then repeat that calculation at each possible point of the spectrogram. We can now calculate a new "spectrogram" by assigning each distance to its respective location.

This process is a sort of convolution. If we estimate k patterns and we apply all of the patterns in the same manner, we will calculate k new spectrograms. From each of the spectrograms we extract the minimum distance at every point in time. In this manner, each spectrogram is reduced to a one dimensional time series. Because the process is applied to all k spectrograms, the result of the algorithm will be a k dimensional time series with the same duration as the original spectrogram.

Figure 10 illustrates this process. We find a representation in which dolphin signals become more prominent and all other sources of sound will be less present. Because we extract the minimum distance at every point in time, we discard the frequency information. In other words, the dolphin signal in the new space could have appeared in any frequency band. In this way the information left is the temporal progression of the signal and the patterns become frequency invariant.
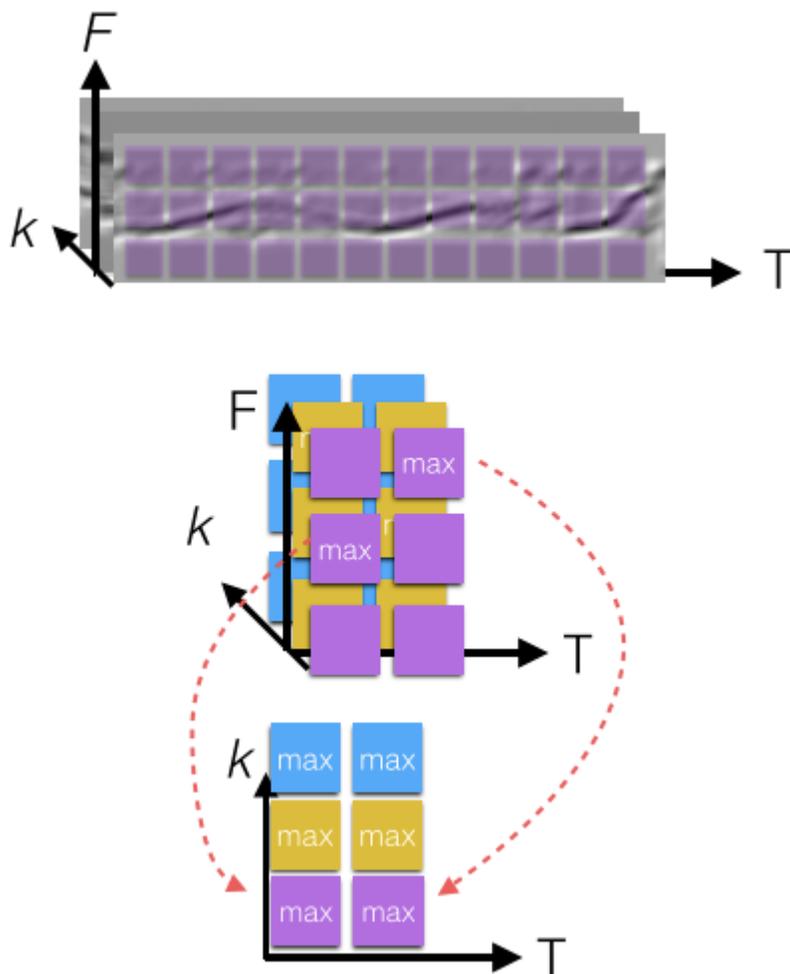
Figure 10. Mapping a sequence into the feature space. At each time slice, each feature patch is compared against the spectrogram at that time over all frequencies (vertical axis). The maximum response is saved for each feature, which results in a k-dimensional series of values through time.

Mining warp-invariant patterns. In this section we 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. We will explain the concept using the dynamic time warping distance (DTW) (Shokoohi-Yekta & Wang, 2015). The second section introduces a probabilistic model for time series called a hidden Markov model (HMM) (Rabiner, 1989). We will also point to divergences and similarities between hidden Markov models and the dynamic time warping distance. The last section introduces an approximate approach to warp-invariant time series comparisons. We will describe the piecewise aggregate approximation and how it can be used as the basis for further discretization.

Dynamic time warping. 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. One solution to this problem is to compute an alignment between the two sequences first. An alignment assigns each sample in X to a close sample in Y (Figure 11). 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. Dynamic time warping is successfully applied to gesture recognition, activity recognition, and speech recognition (Keogh & Ratanamahatana, 2005). Furthermore, the dynamic time warping distance is often used for time series clustering and pattern discovery (Keogh & Ratanamahatana, 2005; Minnen, 2008). In our system, we use the dynamic time warping distance to compare dolphin signals represented in our novel feature space in order to account for time warping effects.
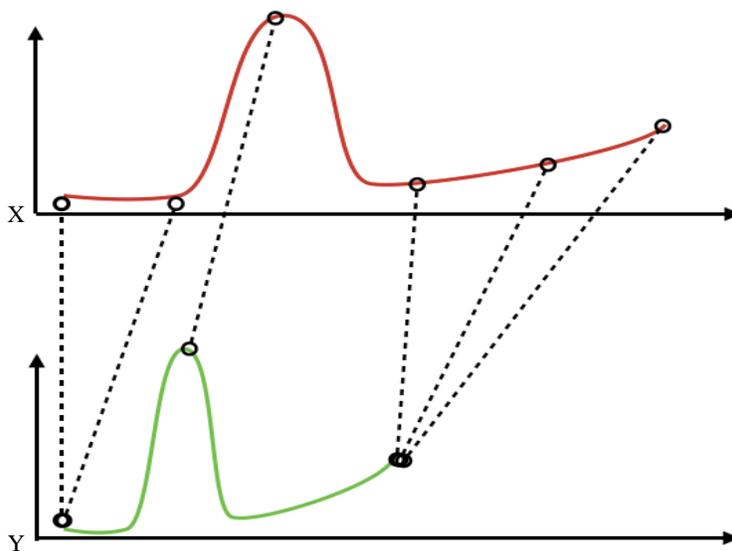
Figure 11. A dynamic time warping example for two hypothetical time series X and Y. The dashed lines indicate which sample from X aligns to which sample in Y.

Hidden Markov models. A hidden Markov model (HMM) is a probabilistic technique that has been successfully applied to various domains such as speech and gesture recognition in the past (Minnen, 2008; Rabiner, 1989). In the following, we will give a general description of hidden Markov models followed by an explanation of how this technique can be used to model a set of dolphin signals and while accounting for time warping effects. We 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, a spectrogram in our feature space is an observation sequence, and each slice of time is an observation. A hidden Markov model is an unobserved Markov chain with k states. The Markov chain is defined by transition probabilities between states. Furthermore, a hidden Markov model is defined by a second function modeling the probability of a sample belonging to a specific state. This probability function is called the observation distribution.
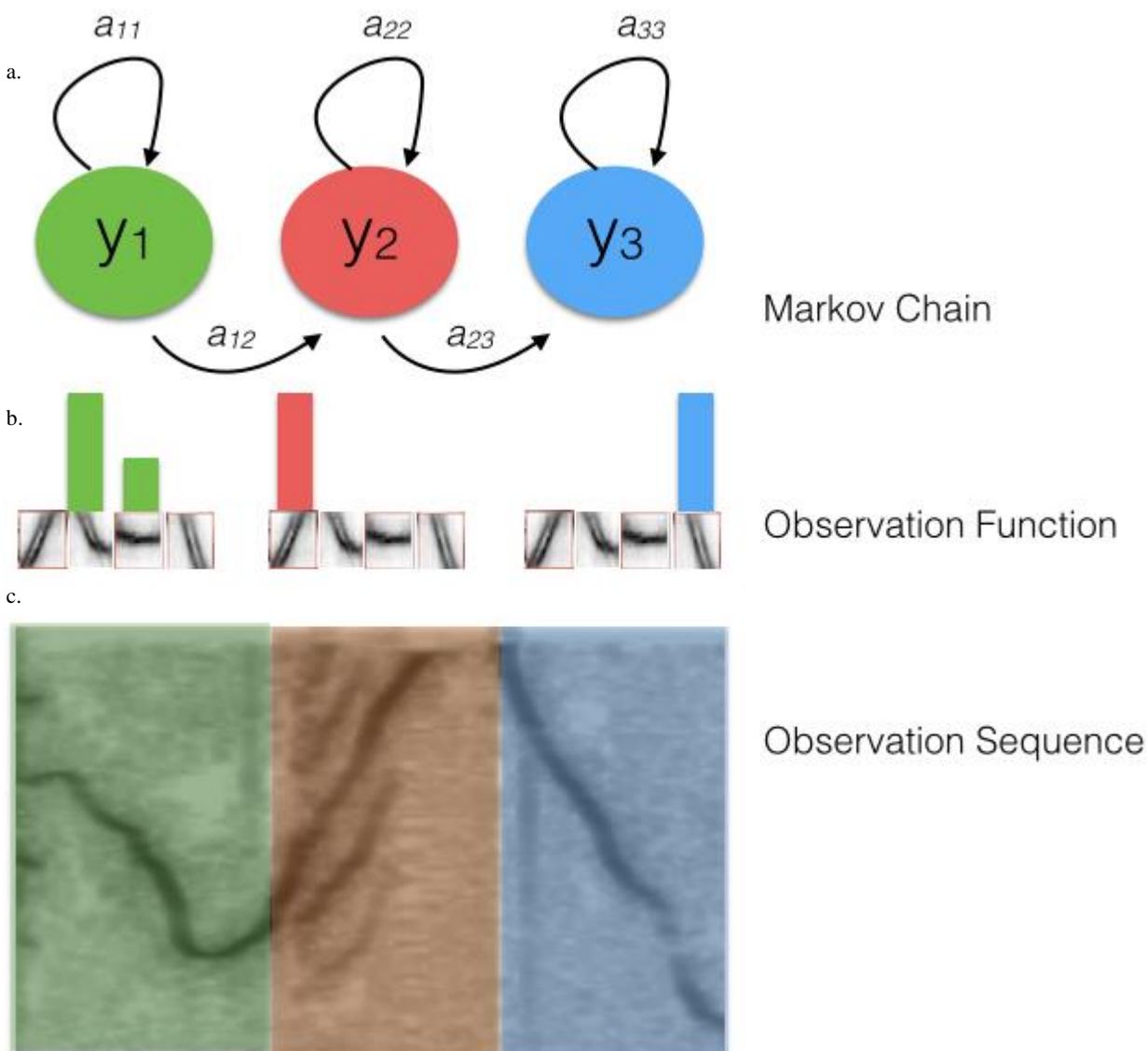
Figure 12. A hidden Markov model with three states. (a) The Markov chain defining the transition function. (b) The observation function for a feature space with four dimensions. Each of the histograms represents a mean vector of a multivariate Gaussian. Each dimension in the mean vector represents the influence of a cluster. (c) 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.

Consider the HMM in Figure 12 as a running example throughout this section. At the top we show the transition function of the HMM as a sparsely connected graph. Each node in the graph represents a state in the HMM. Each edge in the graph represents a non-zero transition probability between states. Often pattern recognition experts model the temporal dynamics of a HMM by defining the connectivity between states. In other words, the experts decide which transitions can have a non-zero probability.

Below the transition function we show the observation function. For continuous multivariate observations, a common observation distribution is a multivariate normal distribution. The transition probabilities and the observation distributions of a HMM can be estimated from a set of observation sequences using an algorithm called Baum-Welch (see Rabiner, 1989). Here, we visualize the mean vectors of the observation distribution, one for each state. In this 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 the order of these states. In our example, we have to transition through the first state before we can transition to 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 spent in each state is coded in the self-transitions, and is equal to the expected value of the geometric distribution. At the bottom of Figure 10, 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 (Rabiner, 1989). 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. 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 HMM 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 becomes higher.

One advantage of using HMMs is that they can be combined into larger ones (Figure 13). For example, if we have two HMMs each modeling a different dolphin signal pattern, we can construct a new HMM 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 HMM for each word. In order to recognize whole sentences, all the word models are combined. The parameters for each word model are trained from multiple audio files for each word.
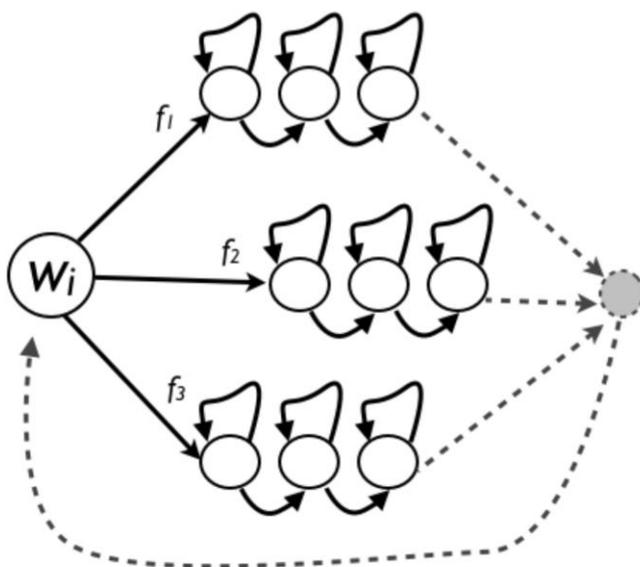
Figure 13. Multiple hidden Markov models combined into a joined model by connecting the end states to each start state.

The discovery process. In this section we describe the complete algorithm for pattern discovery. Leveraging feature invariance and warp invariance, we present an algorithm that takes a spectrogram of dolphin communication as the input and outputs a discrete dolphin communication sequence. Our goal is to learn a representation in which all patterns as well as the underwater noise sources are modeled in a probabilistic model. We learn this model in three steps (Figure 14).

1. Learn the initial segmentation in the novel feature space.
2. Cluster dolphin signals into patterns.
3. Learn a probabilistic joint model of all patterns and noise.
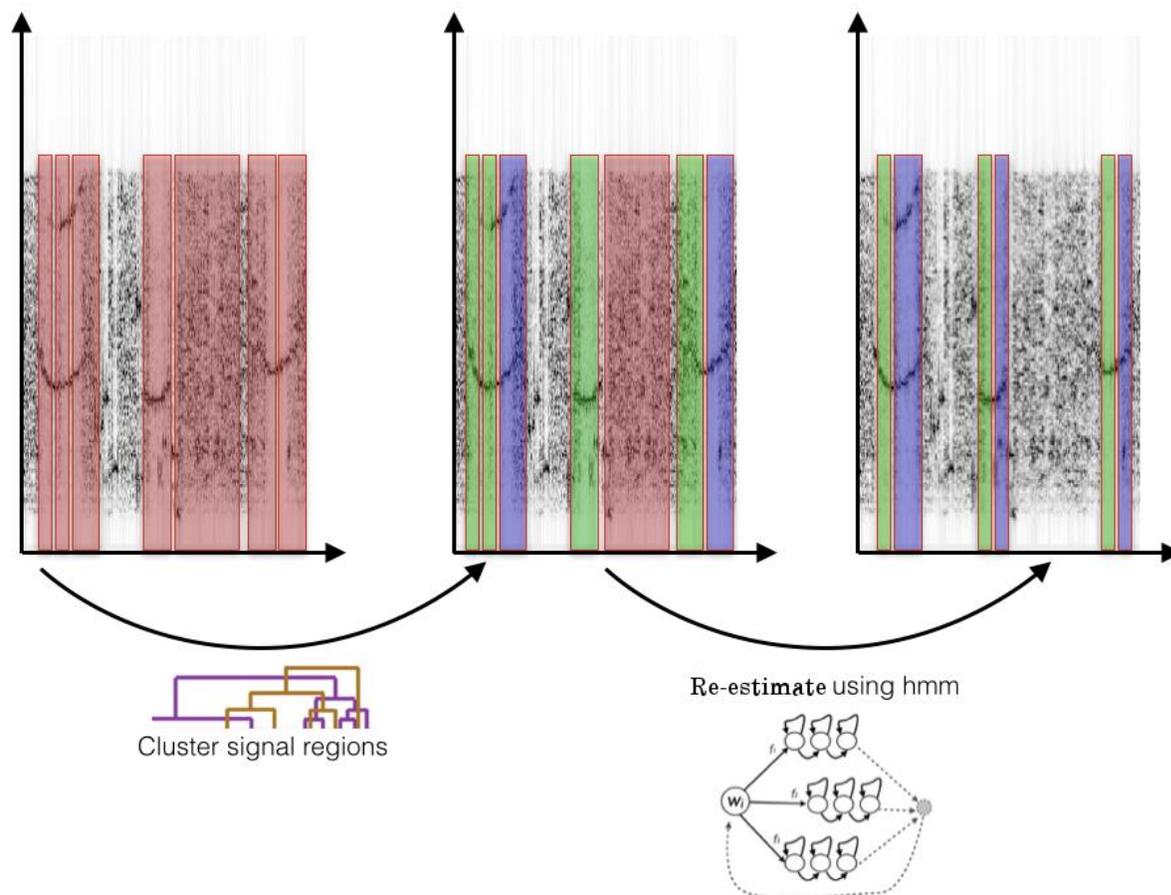
Figure 14. Overview of the data mining system. After identifying signal and noise regions using a binary classifier, the resulting regions are clustered. For the final estimate we train a left-to-right hidden Markov model for each cluster. Together the models form a mixture of hidden Markov models. Decoding each sequence with that mixture gives a smooth segmentation, fixing boundary errors and noise assignments.

In the first step, we convert the spectrogram into the novel feature space. The novel feature space represents the influence of local dolphin signal movement patterns represented as small spectrogram patches. Furthermore, we classify each sample in the sequences as dolphin signal or noise using a random forest (Breiman, 2001). Using the classification results we extract regions of consecutive samples classified as dolphin signal. We extract sliding windows from these regions and cluster the windows into patterns. After the clustering is done, we learn a HMM for each cluster, and we learn a mixture of

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

The resulting HMM can be regarded as the pattern codebook. The model contains every pattern that can be detected in a spectrogram. A final shared pattern codebook is responsible for the conversion of a piece of dolphin communication in the feature space into a dolphin communication sequence.

The algorithm clusters the sliding windows using agglomerative clustering under the dynamic time warping distance (Figure 15). In agglomerative clustering, all windows initially represent their own cluster. In each clustering step, the algorithm merges the two closest clusters. We use the average linkage as the closeness between clusters. Because we are using dynamic time warping, the average linkage between two clusters is the average dynamic time warping distance between all windows. We stop merging at a user-defined maximum number of clusters. We proceed by learning one left-to-right HMM from each cluster using the Baum-Welch algorithm. Because the maximum number of patterns will lead to over-segmentation, we apply greedy mixture learning (Minnen, Isbell, Essa, & Starner, 2007).
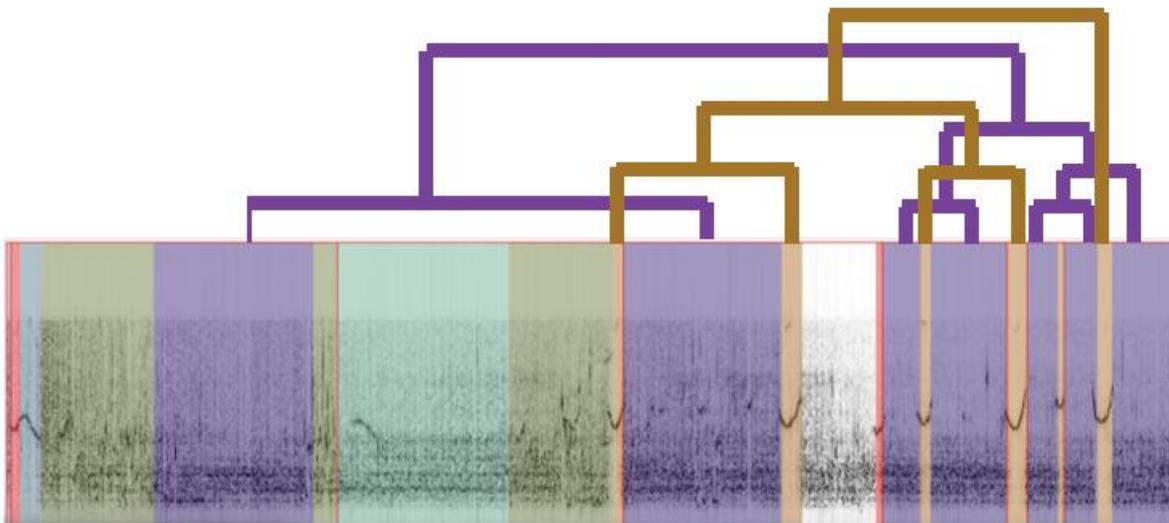
Figure 15. Hierarchical clustering of regions in dolphin communication.

Greedy mixture learning starts with a one-state HMM 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 we can decode all communication sequences in the sequence database using the Viterbi algorithm. By assigning each sample to the pattern indicated by the Viterbi path, we segment the sample into patterns.

Statistical testing in dolphin behavior. In this section, we describe our approach to building statistical models from dolphin communication sequences. Furthermore, we 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. From here on, we assume that the raw audio files were converted into the feature space using automatically learned patches and that all patterns in the audio files of dolphin communication are found using the HMM approach above. In order to run quantitative statistics, we now turn to statistical modeling of pattern distributions in different behavioral contexts.

Our approach to statistical modeling uses discrete distribution estimates as counts of patterns, n-grams of patterns, and sequential rules of patterns in the form of regular expressions. Our communication model is based on distributions of pattern counts, inspired by successful models from information

retrieval (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). In information retrieval, a document is described by word frequencies or the frequencies of features extracted by sentences. In the following, we will describe the statistics we can calculate from a set of dolphin communication sequences. The simplest approach to building the statistics is to build a histogram of pattern occurrences for every dolphin communication sequence. 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 we disregard temporal information. The order in which the patterns occur has no effect in this representation. To be clear, if we used this method with English, the global sentence structure is lost with this representation. The previous statistics model the frequency of patterns or the frequency of local pattern sequences. Our last statistic aims to model the global structure of patterns. We model the global structure of a communication sequence as regular expressions. In Computer Science, a regular expression is a sequence of symbols defining a search pattern. For example the pattern "a (b|c) a [a-z]* c", defines a search pattern that matches strings starting with an "a", followed by a "b" or "c", followed by an "a" again, then a substring of any character and lastly a "c." We automatically extract these search patterns using a technique called "alignment- based learning" (Kohlsdorf, 2015; van Zaanen, 2000). A sample learned regular expression is visualized in Figure 16.
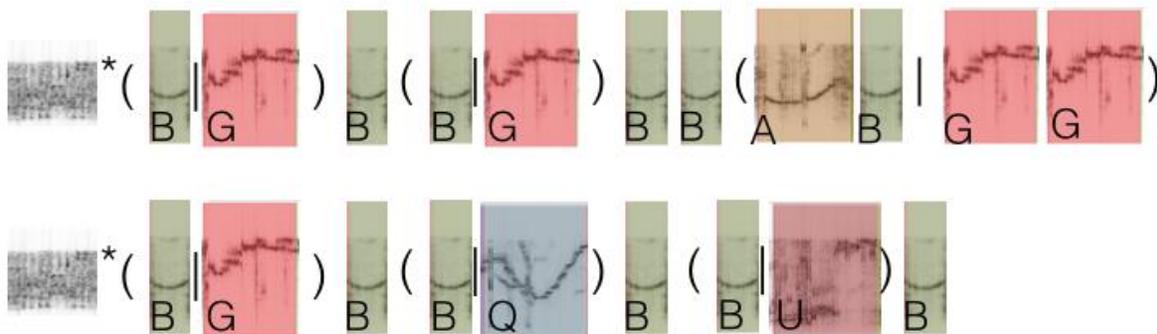
Figure 16. Two rules extracted from our data set. (x|y) represents OR and a star represents a repetition of the previous symbol as often as needed to match the rule.

Evaluation of the patterns found. The following experiments evaluate the statistical testing capabilities of our method. We use a dataset to run comparative statistics among different behavioral contexts. Audio files from the Wild Dolphin Project are annotated with the observed behavior contexts in which they were collected: play behavior, foraging behavior, aggressive behavior and mother- calf reunions. In total, the dataset contains 25 audio files: seven files showing aggressive behavior, five files with foraging behavior, six files with play behavior and seven files with data from mother-calf reunions. Our marine mammalogist selected these contexts because the 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. We 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 we use two datasets. The first dataset contains the four behavioral contexts. The second dataset contains the four contexts combined with other patterns from the Wild Dolphin Project's 2012 field season and contains 67 audio files of unannotated data. Furthermore, we use

a dataset 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 three noise files, 78 whistles, and 15 burst pulse examples. From the whistles and burst pulses, we automatically learn our 60 dimensional feature space (as described in Figure 1) and convert the four-context dataset as well as the unannotated files into the feature space. We use the noise data to learn a signal detector. Then we run the pattern discovery algorithm and extract all the patterns with two conditions:

1. Discovery in four contexts only.

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

We do so in order to evaluate if adding more, unannotated data, will stabilize the pattern estimation. Our hypothesis is that the more data we have in the discovery stage, the better the patterns will be. In the last step we calculate all statistics: the simple pattern count, the n-gram count and the rule count into a single statistic per audio file. The combined statistic is simply all counts concatenated in a large histogram. Some patterns found in the large dataset are shown in Figure 17.

We then run comparative statistics in each condition and perform comparisons between two contexts using a Chi-square test on the histogram statistics extracted from each dolphin communication sequence. When comparing data from context c1 and context c2:

1. Estimate a distribution c1, and test if c2 is from that distribution.

2. Estimate a distribution c2, and test if c1 is from that distribution.

The method returns two p-values, one for each case. These *p*-values are used to indicate a significant difference between the contexts. 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. We 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, we apply Bonferroni correction. The Bonferroni correction accounts for the family-wise 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 were available for these initial tests.
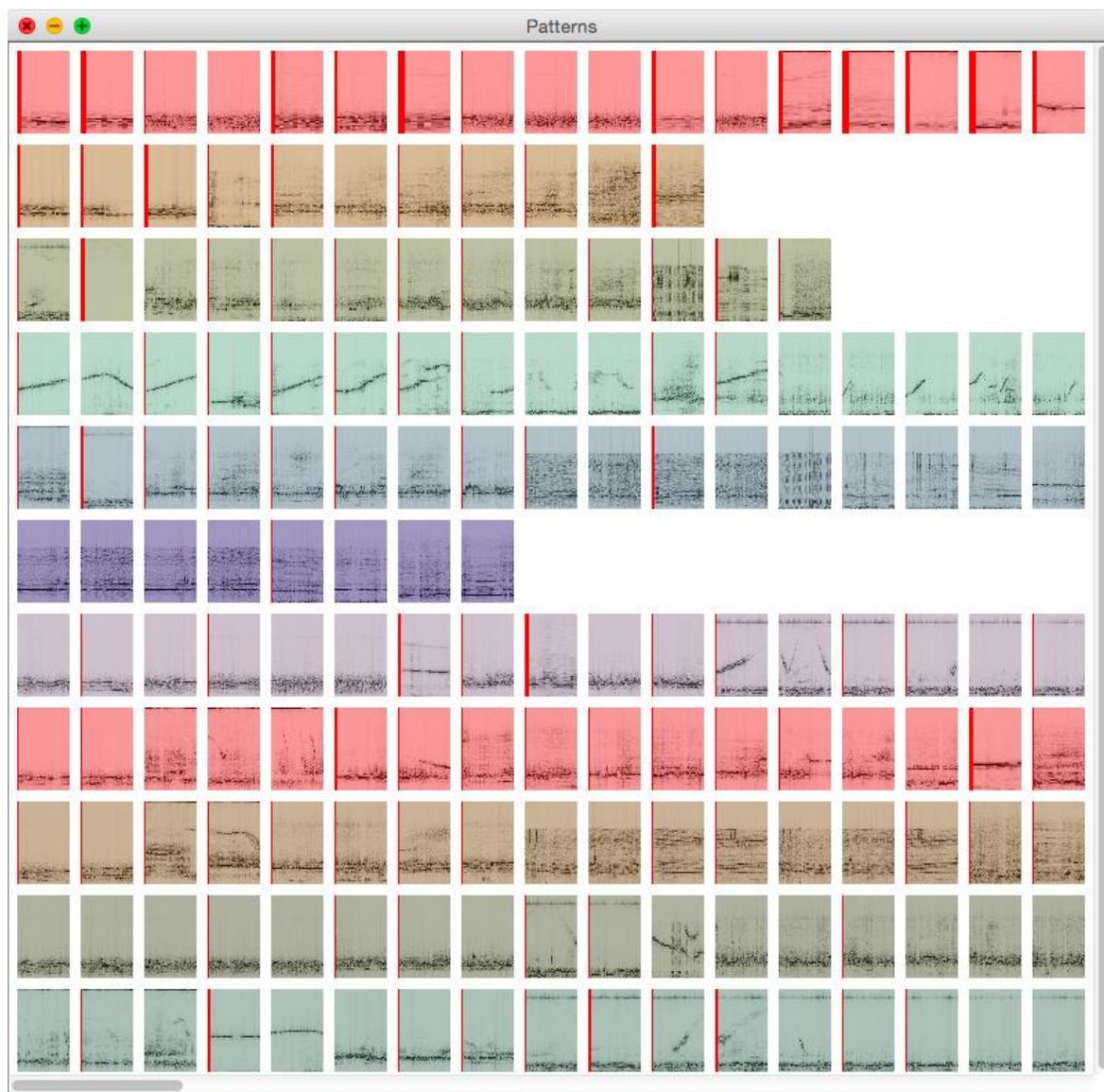
Figure 17. The pattern view in the signal imager for the exact discovery in the large dataset. Each row represents one pattern. Each column represents an example of that pattern. All examples are color-coded indicating the pattern.

When comparing data from a context to itself, we split the data into two random subsets of equal size and run the tests between the two. We 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 1. 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 of 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, we observe an unexpected result. It seems that foraging behavior and play behavior are not significantly different. However, these results support that the communication during foraging behavior is characterized by echolocation and communication during play is more often characterized by whistles.

Table 1

The p-values for the statistical testing experiment using the small dataset and the exact algorithm

	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

Note. Significant *p*-values after correction are shown in green. Non-significant values are shown in blue.

Table 2

The p-values for the statistical testing experiment using the combined dataset and the exact algorithm

	Aggression	Play	Foraging	Reunion
Aggression	0.51	$7.48e^{-11}$	0.02	$1.45e^{-13}$
Play	$< e^{-14}$	0.79	$1.46e^{-14}$	0.01
Foraging	$< e^{-14}$	$< e^{-14}$	0.98	$1.63e^{-7}$
Reunion	$3.00e^{-9}$	$2.58e^{-14}$	$< e^{-14}$	0.84

Note. 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.

In the second experiment, we combine data from the 2012 dataset with the four- context dataset. Our hypothesis is that a larger set of data will stabilize the pattern discovery and, in turn, provide cleaner statistics. We perform the same statistical testing as in the first experiment. The results of the second experiment are shown in Table 2. As one can see, the combined dataset shows higher *p*-values for the diagonal and lower p values in every other cell. The unexpected result in the combined condition occurs only after correction for multiple tests. These results suggest that, on average, our discovery algorithm, when run on acoustic data, does produce different patterns for the behavioral contexts than those determined visually by experts.

Conclusion

These experiments, using dolphin vocalization samples and behavioral labeling, show that this system is able to perform comparative statistics on auditory data from several contexts. We 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 shows strong similarity. 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 result in these experiments: adding more data to the pattern discovery process results in better statistics. Our intuition is that a large dataset is desirable for pattern discovery. The parameters of HMMs become more stable when estimated from more data. A more stable estimate of the HMMs' parameters results in a better decoding of sequences and, in turn, a better estimate of the statistics. In on-going work, we are collecting larger datasets using multiple field seasons from the Wild Dolphin Project to further test our pattern discovery algorithm. We are also focusing on designing our Signal Imager tool to allow easy visualization of the discovered patterns and statistics in large datasets. Our hope is that such tools will help researchers mine and categorize information much more efficiently.

References

Adi, K., Sonstrom, K., Scheifele, P., & Johnson, M. (2008). Unsupervised validity measures for vocalization clustering. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4377–4380.

Ashbrook, D. & Starner, T. (2003). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing, 7,* 275–286.

Baggenstoss, P. M., & Kurth, F. (2014). Comparing shift-autocorrelation with cepstrum for detection of burst pulses in impulsive noise. *The Journal of the Acoustical Society of America, 136*, 1574–1582.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Coates, A., Lee, H., & Ng, A. Y. (2011). An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, *1001,* 215–223.

Deecke, V. B., Ford, J. K., & Spong, P. (1999). Quantifying complex patterns of bioacoustic variation: Use of a neural network to compare killer whale (*Orcinus orca*) dialects. *The Journal of the Acoustical Society of America, 105*, 2499–2507.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*, 391-407.

Esfahanian, M., Zhuang, H., & Erdol, N. (2014). On contour-based classification of dolphin whistles by type. *Applied Acoustics, 76,* 274–279.

Halkias, X., & Ellis, D. (2006). Call detection and extraction using Bayesian inference. *Applied Acoustics, 67*, 1164–1174.

Harley, H. E. (2008). Whistle discrimination and categorization by the Atlantic bottlenose dolphin (*Tursiops truncatus*): A review of the signature whistle framework and a perceptual test. *Behavioural Processes*, *77*, 243–268.

Herzing, D. L. (1996). Vocalizations and associated underwater behavior of free-ranging Atlantic spotted dolphins, *Stenella frontalis* and bottlenose dolphins, *Tursiops truncatus*. *Aquatic Mammals, 22*, 61–79.

Herzing, D. L. (2000). Acoustics and social behavior of wild dolphins: Implications for a sound society. In W. W. L. Au, A. N. Popper, & R. R. Fay (Eds.), *Hearing in whales, Springer-Verlag handbook of auditory research* (pp. 225–272). New York, NY: Springer-Verlag.

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems, 7*, 358-386.

Kershenbaum, A., Sayigh, L. S., & Janik, V. M. (2013). The encoding of individual identity in dolphin signature whistles: How much information is needed? *PLoS ONE, 8*, e77671.

King, S. L., Sayigh, L. S., Wells, R. S., Fellner, W., & Janik, V. M. (2013). Vocal copying of individually distinctive signature whistles in bottlenose dolphins. *Proceedings of the Royal Society B: Biological Sciences, 280*, 20130053. http://dx.doi.org/10.1098/rspb.2013.0053.

Kohlsdorf, D. (2015). *Data mining in large collections of dolphin signals* (Unpublished thesis). Georgia Institute of Technology, Atlanta, GA.

Kohlsdorf, D., Mason, C., Herzing, D., & Starner, T. (2014). Probabilistic extraction and discovery of fundamental units in dolphin whistles. In *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 8242–8246.

Lampert, T. A., & O'Keefe, S. E. (2010a). A survey of spectrogram track detection algorithms. *Applied Acoustics, 71*, 87–100.

Lampert, T. A., & O'Keefe, S. E. (2010b). An active contour algorithm for spectrogram track detection. *Pattern Recognition Letters, 31*, 1201–1206.

Minnen, D. (2008). *Unsupervised discovery of activity primitives from multivariate sensor data*. PhD dissertation, College of Computing, Georgia Institute of Technology, Atlanta, GA.

Minnen, D., Isbell, C. L., Essa, I., & Starner, T. (2007). Discovering multivariate motifs using subsequence density estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 615–620.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286.

Richards, D. G., Wolz, J. P., & Herman, L. M. (1984). Vocal mimicry of computer-generated sounds and vocal labeling of objects by a bottlenosed dolphin *Tursiops truncatus*. *Journal of Comparative Psychology, 98*, 10–28.

Sayigh, L. S., Esch, H. C., Wells, R. S., & Janik, V. M. (2007). Facts about signature whistles of bottlenose dolphins, *Tursiops truncatus*, *Animal Behaviour, 74,* 1631–1642.

Shapiro, A. D., & Wang, C. (2009). A versatile pitch tracking algorithm: From human speech to killer whale vocalizations. *The Journal of the Acoustical Society of America, 126*, 451–459.

Shokoohi-Yekta, M., Wang, J., & Keogh, E. (2015). On the non-trivial generalization of dynamic time warping to the multi-dimensional case. *Proceedings of the SIAM International Conference on Data Mining*, 39–48.

Slobodchikoff, C., & Placer, J. (2006). Acoustic structures in the alarm calls of Gunnison's prairie dogs. *Journal of the Acoustical Society of America, 119*, 3153–3160.

van Zaanen, M. (2000). ABL: Alignment-based learning. *Proceedings of the 18th International Conference on Computational Linguistics*, 961–967.

Zakaria, J., Rotschafer, S., Mueen, A., Razak, K., & Keogh, E. (2012). Mining massive archives of mice sounds with symbolized representations. *Proceedings of the International Conference on Data Mining*, 588–599.