Feature Learning and Automatic Segmentation for Dolphin Communication Analysis

Anonym

[1] Anonym

author@university.edu, coauthor@company.com

Abstract

The study of dolphin cognition involves intensive research of animal vocalizations recorded in the field. We address the automated analysis of audible dolphin communication and propose a system 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. Our results indicate that our system provides meaningful patterns to the marine biologist and that the comparative statistics are aligned with the biologists domain knowledge.

Index Terms: Pattern Discovery, Bio Acoustics, Marine Mammals, Dolphin Behavior

1. Introduction

Dolphin cognition and communication research is a significant subfield 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 often 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.

We propose a feature learning algorithm and a pattern discovery algorithm for audible dolphin communication. Furthermore, we use the resulting patterns to enable marine mammalogists to perform statistical tests in order to reason about the differences in dolphin communication depending on the communication's context. The feature learning algorithm is based on convolutional features extracted from the spectrogram and the pattern discovery algorithm is based on hidden Markov models. Furthermore, we describe an algorithm that learns regular expressions from pattern sequences using alignment based learning. In an experiment we show that the resulting patterns (see Figure 1) are indeed usable by a domain expert to run statistical tests between communication observations collected in different contexts.

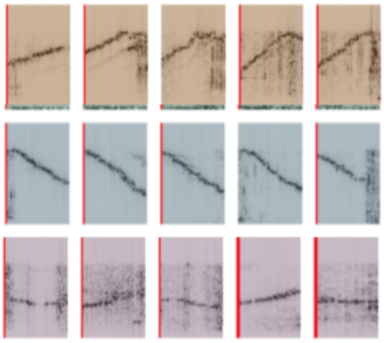

Figure 1: Multiple Patterns extracted from spectrograms during our experiment. Each row represents one pattern. Each column represents an example spectrogram snippet of that pattern. All examples are color coded indicating the pattern.

2. Related Work

In a survey on underwater acoustics processing methods, Lampert and O'Keefe [1] identify three main algorithm categories: image processing, neural networks, and statistical models. They evaluate several methods on a dolphin whistle detection task. 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. [2] 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 [3], who compare methods for detecting burst pulses in impulse noise, and Kohlsdorf et al. [4], who trace a dolphin whistle using a probabilistic pitch tracker. Other approaches to whistle extraction include a frame-based Bayesian approach [5] and a Kalman filtering approach [6]. Shapiro and Wang apply pitch detection designed for human telephone speech to whale vocalizations [7].

Dolphin signal clustering and classification often uses neural networks [8, 9] or clustering based on hidden Markov models [10]. Both approaches filter the data first and use Mel-cepstral coefficients or other measurements from the spectrogram as features. Our goal is to improve the efficiency and efficacy of analysis of clicks, whistles, and bursts [11]. We adopt an approach similar to the work done by Zakaria et al. on mining archives of mouse sound using symbolic representations [12].

3. Dolphin Communication Mining

Biology researchers capture audible dolphin communication in digital field recordings. We propose a system that automatically learns a feature representation from a spectrogram and describe how to use hidden Markov models and hierarchical clustering to discover short dolphin signal patterns in this novel feature space. Furthermore, we describe an algorithm that uses alignments to model sequences of these patterns.

Our approach has parallels to speech recognition where the acoustic signal is modeled, then a phoneme recognizer is trained, and finally sequences of phonemes are modeled with contexts and grammars. As in a speech recognition system we begin modeling the signals in the spectrogram of the audio measurements. Here we define a spectrogram 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. We define a signal pattern as a set of subsequences from several spectrograms that appear similar to each other given a distance function. Our system uses the feature learning and hidden Markov models to convert the spectrogram into 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.

3.1. Feature Learning For Dolphin Communication

In order to enable frequency-invariant comparison of dolphin signals, we learn a set of k feature extractors spanning a k-dimensional feature space. Using these features, 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 easily distinguish between dolphin signals and other underwater noise sources.

Using these feature extractors we 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 [13] 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.

We learn the feature extractors from a dataset of audio files that are categorized into dolphin whistles, burst pulses or noise. 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 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 our experiments a patch represents approximately half a millisecond in time and one kHz in frequency.

Given the spectrograms extracted from the catalog, we extract all patches that fall around dolphin communication. For all whistles, we use a whistle tracer [4] and use only patches along the whistles trace. For burst pulse like signals, we extract interest points (a point of high magnitude that is maximal in a local neighborhood) in the spectrogram first [14] and use only patches around these interest points.

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 [13] to form a codebook of regions. We z-normalize each region before proceeding [13].

We 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 2.

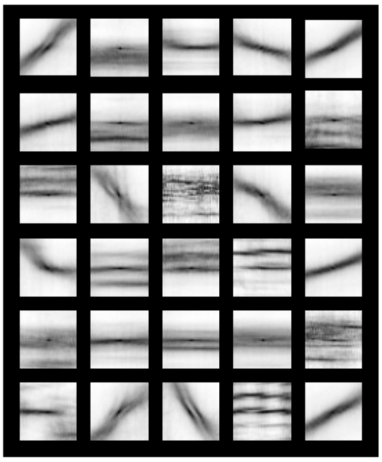

Figure 2: A set of 30 feature extractors learned using k-means.

The resulting codebook represents our feature extractors. A cluster is a square region c with length $2d$ and mean mu. In order to transform a spectrogram into the new feature space spanned by the codebook, we perform the following steps. We 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 we shift the region over the spectrogram and replace each spectrogram point with the distance, we get a new time series $S^c = \{s^c_1 ... s^c_t\}$. Each point s^c_{tf} in the new sequence represents the distance of the region c to the spectrogram around the point s_{tf}:

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

We 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^{ci}_{tf} represents the distance of the spectrogram area centered at time t and frequency f to the cluster center c_i. Next we transform the distance representation in a representation capturing the response or influence of each cluster. First we compute the mean of all k distances at every point in the spectrogram: The influence of a cluster at a point in time t and frequency is

$$s^c_{tf} = max(0, \mu_{tf} - s^c_{tf}) \tag{4}$$

Now each point s^c_{tf} 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 [13]. 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. The complete process is shown in Figure 3. As one can see on the top, we 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.

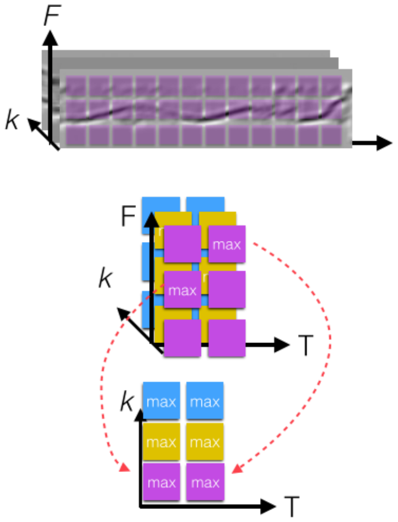

Figure 3: Mapping sequence into feature space.

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}.

3.2. Pattern Discovery

With the novel feature space we learned an acoustic model for dolphin communication. In the next step, we use hidden Markov models to build a probabilistic model of dolphin patterns. In particular we 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 $P = \{p_1, ...p_T\}$.

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. In the first step, we convert the spectrogram $S = \{s_1...s_T\}$ into the novel feature space described in the previous section $S' = \{s'_1..s'_T\}$. Furthermore, we classify each sample in the sequences as dolphin signal or noise using a random forest [15]. 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 hidden Markov model for each cluster. Furthermore, we learn a mixture of Gaussian from the samples classified as noise. We then combine the resulting models and the mixture into a joint 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 $P = \{p_1..p_N\}$.

The clustering algorithm clusters the sliding windows using agglomerative clustering under the dynamic time warping distance. In agglomerative clustering, all windows initially represent their own cluster. In each clustering step, the algorithm merges the two closest clusters under dynamic time warping distance. We use the average linkage criteria as the closeness between clusters. We 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, we apply greedy mixture learning [4, 16].

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 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 achieve a segmentation into patterns: $P = \{p_1, ...p_T\}, p_i \in \{1...N\}$.

3.3. Pattern Rules

Now that we can build a probabilistic model of dolphin sequences, we can turn to modeling sequences of these patterns. In particular, we describe an algorithm that learns a set of regular expressions from unlabeled dolphin signal sequences converted into the pattern representation. This representation can later be used to gather statistics about dolphin communication.

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". In the following, we will describe how to learn a set of regular expressions from a database of dolphin communication patterns using an algorithm called "alignment-based learning" [17]. The resulting regular expressions support regions where no character matches and regions with an OR.

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. We use the Needlemann-Wunsh algorithm to construct the alignments [18]. From the alignment, we can retrieve the insertions, deletions and match operations. All regions of matches are unchanged. We 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 $(abc|def)$. Now we build a set of all regular expressions from the dataset. Then, we align each sequence to each other sequence and extract the regular expression. We add a regular expression to the set of regular expressions if it matches more sequences than a predefined threshold. In Figure 4 we visualize some of the regular expressions extracted for our experiments.

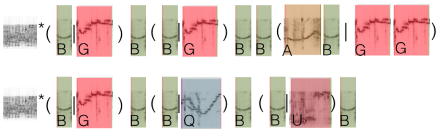

Figure 4: Two rules extracted from our data set. $(x|y)$ represents OR and $*$ represents a repetition of the previous symbol as often as needed to match the rule.

4. Statistical Testing Experiments

We use a dataset of dolphin communication sequences 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. We augment the dataset with 65 unlabeled sequences and also gather a small dataset of annotated dolphin signals to learn the feature space.

Using the database, we learn a feature space with 60 dimensions. We build a random forest with 10 trees to filter the noise as described above. We then learn a mixture of three state left-right hidden Markov models and learn a set of regular expressions from all pattern sequences.

In order to run comparative statistics between different contexts, we convert each audio file in the database into its pattern representation and then build a histogram that combines the occurrence count of each pattern in a context as well as the matching count of each regular expression in each context. In other words, we have one histogram for aggressive behavior, one histogram for foraging behavior, one for play behavior and one for mother calf reunions.

In particular we run comparative statistics in each condition and perform comparisons between two contexts' histograms using a χ^2 test on the histogram statistics extracted from each dolphin communication sequence.

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 our experiments, we use a 0.95 confidence interval.

Table 1: The p-values for the statistical testing experiment using the combined dataset. Significant p-values after correction are shown in green. Non-significant values are shown in blue. Values that are non-significant after Bonferroni 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

When comparing data from a context to itself, we split the data into two equally sized subsets and run the tests between the two and repeat the testing process 10 times and average the resulting *p-values*

It is worth mentioning that the system is trained on a larger portion of the data then used for the statistical analysis. The pattern sequences and hidden Markov models are all estimated using all the data combined, the unlabeled sequences as well as the annotated sequences with the context labels. We chose to follow this route, since we observed unstable model estimates when using small datasets. The numerics as well as the model's quality with respect to the statistical testing improved with more data.

5. Discussion

As one can see in Table 1, we observe non significant values along the diagonal of our testing matrix and significant values in close to all fields. In other words, there is no observable difference when comparing a context to itself; the pattern distribution is similar within a context. Furthermore, we observe noticeable differences in the off diagonal fields which indicates that the pattern distribution is not similar across contexts. For example, we observe a statistical difference of communication between aggression and mother calf reunions. However, there is no difference when comparing different sets of audio files containing aggressive behavior. These results indicate that the acoustic patterns found match the visual distinctions made by biologists when defining these categories.

6. Conclusion

We presented an algorithm that uses feature learning, hidden Markov models and alignment based learning to produce models of dolphin communication patterns sequences. Furthermore, in an experiment we presented results that indicate that the learned models can be used to run quantitative experiments using statistical testing. These models can help biologists in the future, to quickly produce models of dolphin communication that can be used to validate hypothesis about dolphin communication and behavior in different social contexts.

7. References

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

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

[3] P. M. Baggenstoss and F. Kurth, "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.

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

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

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

[7] A. D. Shapiro and C. Wang, "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.

[8] V. B. Deecke, J. K. B. Ford, and P. Spong, "Quantifying complex patterns of bioacoustic variation: Use of a neural network to compare killer whale (*Orcinus orca*) dialects," *Journal of the Acoustical Society of America*, vol. 105, no. 4, pp. 2499–2507, 1999.

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

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

[11] D. Herzing, "Acoustics and social behavior of wild dolphins: Implications for a sound society," *Hearing in Whales, Springer-Verlag Handbook of Auditory Research*, 2000.

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

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

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

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

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

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

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