UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de Doctorat:

AUTOMÀTICA, ROBÒTICA I VISIÓ

Tesi Doctoral

AN ACOUSTIC BIO-METRIC FOR SPERM WHALES

Mike van der Schaar

Director: Michel André
Codirector: Andreu Català

Octubre de 2009

Contents

List of Figures

List of Tables

Chapter 1

Summary

Acoustic identification of sperm whales, or sperm whale groups, has both a biological and practical use. The biological interest is especially in the ability to separate recordings with mixed vocalisations to reconstruct the individual time series. This allows to follow single animals during their dive and to track their activity. Practically, for passive detection, localisation and monitoring applications it is often useful to have an estimate on the number of animals in an area. Additionally, the ability to identify which signal belongs to which animal can aid and speed-up both tracking and localisation.

This thesis presents an approach for identification of sperm whales using acoustic cues with the requirement that the algorithm can run on limited resources in real-time. First, an attempt was made to identify sperm whale groups using the rhythmic structure in codas. A protocol was developed for objective coda classification and to compare results between studies. Due to a lack of data it is not yet clear if the coda rhythm is truly unique for a sperm whale group. To further investigate this a software tool has been developed and made available to the research community. If adopted, this could allow to draw better conclusions concerning the importance of the rhythm in codas in the future.

After codas were discarded for identification, attention was focussed on finding characteristic information in the time-frequency domain of a sperm whale click. The Gabor function was first used as a model for a click, but was found to be too limited in its choice for characteristic features and too unreliable due to variability in the click itself. To generate a large selection of possible features a local discriminant basis was created based on a wavelet packet table. This led to features that outperformed the Gabor function with a simple linear classifier.

Studying the feature space constructed from the wavelet coefficients suggested the use of a non-linear classifier that can model the features' clusters. Using Gaussian kernels to describe the clusters, a radial basis function network (RBF) was constructed for the classification. Additionally, besides the use of the RBF network that focussed on the cluster centres, classification with support vector machines (SVM), which focus on the cluster boundaries, was also evaluated. It was found that a model describing cluster centres with RBF outperformed SVM. To its advantage, the RBF network required much less information and computed much faster.

To initialise the classifiers a Gaussian Mixture Model (GMM) clustering algorithm was evaluated. The task of this algorithm is to perform an initial separation, using the Gaussian distributions that showed good performance with RBF, that allows the classifier to be trained. The optimisation of GMM makes use of an expectation maximisation routine, a much more expensive algorithm than that was used for RBF. Therefore, it is intended to only run at the start of a recording to obtain a training set, after which RBF can take over.

Clustering with GMM showed capacity to estimate the number of animals and to provide enough information to train the RBF classifier, but this should be tested on more data sets in the future.

The classification approach herein presented allowed accurate classification on the available data set in real-time, and the approach is considered to be a reliable method that can be applied in autonomous monitoring applications or e.g. from a laptop on a boat.

Chapter 2

Acknowledgements

From the university of Delft I would first like to thank Dr. Cees Kamminga who introduced me to the analysis of sperm whale clicks in 1998 during a 3-month practical assignment, and who afterwards offered me the opportunity to continue that work 2001 for six months together with Dr. Hans van der Weide.

I want to thank Dr. Michel André, for directing and his support on the thesis, and for providing the opportunity to have performed this research.

I want to thank Dr. Andreu Català for accepting co-directorship on the thesis and his comments on various parts of the work.

I want to thank the reviewers Dr. Cedric Gervaise and Prof. Manell Zakharia for their comments and improvements on this report.

I want to thank Eric Delory for the fruitful collaboration both at the UICMM at the university of Las Palmas, ands at the LAB at the UPC.

I want to thank Dr. Natalie Jaquet for providing the vital data recordings.

I want to thank the crew at the LAB for providing an amiable and productive working environment.

And I want to thank my family for their continuing support and enthusiasm after I moved to Spain and their willingness to visit me at the various beaches of Las Palmas and Barcelona.

Part of the study was funded under a grant of the BBVA (Banco Bilbao Vizcaya Argentaria) Foundation, and a research grant from the UPC.

Chapter 3

Introduction

3.1 Practical relevance of the thesis

At the moment, acoustic identification of individual marine mammals (e.g. dolphins, sperm whales) is scarcely done. Identification is mainly performed based on photo identification of dorsal fins (see for example [41, 59, 103]). This type of visual identification is very effective during boat surveys with staff available to take pictures of animals that surface frequently. This approach is impossible for an autonomous system installed at sea or for animals that can stay submerged for long periods of time. In these cases using acoustics would be much more appropriate for identification, as most marine mammals produce sound for echo location or communication. As explained in more detail in Section 4.1, there has been work on separating signals from different species and classifying specific calls from a single species (whistles from dolphins or codas from sperm whales), but there has not been much success in recognising or separating individual sperm whales in a recording for real-time applications, using purely acoustic cues. Here, the importance of a real-time algorithm is not just in computational time of the signal processing, but also in the limitation of available data. The classifier will need to process each click directly as it arrives and cannot wait to collect a few more signals to reach a decision. As will be shown in Chapter 8, the classification problem is more difficult when it can only be trained on a small part of the start of a recording than when it could make use of all available data (as with e.g. off-line clustering techniques) due to variability of the features.

Initially, the presented work was carried out to be applied to the Whale Anti-Collision Project (WACS). This project is primarily aimed at reducing the number of collisions between shipping traffic and sperm whales in the channel between Gran Canaria and Tenerife (Canary Islands). These types of collisions have become a concern as the mortality rates are increasing [52]. The proposed system is shown in Figure 3.1; it consists of a series of detection buoys separated by 10 km that are capable of detecting actively vocalising whales through time delay of arrival techniques at the hydrophones at the buoy and silent whales through ambient noise imaging techniques [79, 80]. The system is passive to minimise the influence of its presence to the environment. These autonomous buoys would broadcast the information they gather to a shore station, where it is combined and transmitted to ships in the area. This region is known for its resident sperm whales [5] that often forage in groups. While there is no danger of collisions when the animals are foraging, for tracking purposes it is important to have an idea about the number of animals that are under water and to follow them during the dive. The algorithms for identification and separation of sperm whales were designed to run at the shore station for real-time post processing of the incoming data.

While the WACS system is still an important research objective, funding has been difficult to obtain as especially ambient noise imaging requires high-end hardware and both the development and deployment is

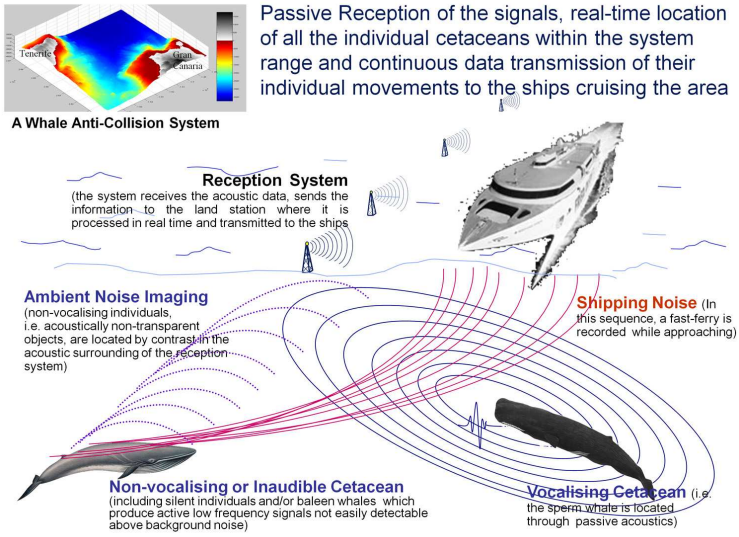

Figure 3.1: Schematic of the WACS system. Multiple buoys are deployed to cover a channel where vocalising whales are detected directly. Passive whales are detected through ambient noise imaging. Results are sent to shore and transmitted to ships in the area.

quite expensive. Nevertheless, over the last years passive monitoring has become a major research topic. This has been pushed by two factors. First, due to the dangers of tsunamis, concerns about climate change and other research objectives, especially neutrino detection, many underwater research and monitoring platforms are being realised around the coasts of e.g. Europe (ESONET [75], ANTARES [2]), United States (MARS [58]), Canada (NEPTUNE [76], VENUS [29]) and Japan (DONET [10]), while other networks are under consideration. These platforms are generally set up with equipment aimed at monitoring the chemistry and salinity in the water column and detectors for seismic activity.

Second, recently noise has been recognised as a source of pollution to the environment, including the under water environment, and it has been listed as such in the Marine Strategy Framework Directive of the European Union [35]. In order to help regulating the noise production at sea, the LAB has created a best practices document for the Spanish government under the eCREM project (contract number 083/SGTB/2007/1.4). One example where noise is a large concern is construction at sea (e.g. pile driving), especially that of windmill farms, which typically produces high intensity impulsive noise [17, 27]. It has already been argued that this can affect or kill fish [78], and it is currently unknown what the immediate effects are on marine mammals and the environment in general. In order to find out how noise influences the underwater environment it is important to design monitoring systems equipped with hydrophones that cannot only monitor the sound exposure levels but detect presence of marine mammals as well.

The LAB is currently developing such a system called LIDO (listening to the deep ocean environment, http://lido.epsevg.upc.es) where existing platforms can be equipped with hydrophones and the necessary detection software. The project is being realised as a demonstration mission within the ESONET (European Commission Framework Program 6, contract number 036851) and will be implemented at least at the NEMO test site. As a member of the ANTARES consortium, the LAB will implement the same system at their platform, and currently a project is being proposed to do the same with VENUS in Canada. The general LIDO design is shown in Figure 3.2 and consists of three stages. The first stage detects acoustic events like

impulses, whistles, or constant tonal sounds. The detection of these events is intended to be done close to the source of the data stream (hydrophones). This allows to minimise the amount of data that needs to be sent to shore when only limited bandwidth is available, or when there is no communication line to shore it minimises the data that needs to be stored. The second stage processing handles classification, localisation and detection of the acoustic events. These first two stages operate in real-time and the results can be viewed by the general public using a client that connects to the data streams that are offered from the platform sites. These streams include a single compressed audio channel and all the analysis results. Third stage processing is done off-line and its purpose is to analyse the data over a longer time period to find correlation between noise levels and e.g. cetacean presence or other trends.

Depending on the purpose of the system, identification and separation of sperm whales will be done at the first or second LIDO stage. The first stage is considered when the platform is moored and uncabled and there is an interest in especially tracking sperm whales at the location. In this case it is essential that the algorithms run fast with minimal hardware requirements in order to save power. Otherwise, the software will run at the second stage on shore and power consumption of the system is less of a concern. When only one hydrophone is available the only method available to differentiate between the animals and follow them during the dive is through acoustic cues in their signals. When multiple hydrophones are available the acoustic identification will work together with other tracking algorithms based on time delay of arrival at the hydrophones. It is noted that there has been some research into using a single hydrophone for source localisation [93, 53], but these approaches require intensive manual supervision to identify a source signal and its surface or bottom reflections, in combination with assumptions on the environment that are not always realistic (e.g. extremely calm seas). It is very difficult to automate these algorithms and therefore we rely rather on acoustic cues for identification in single hydrophone situations then localisation techniques.

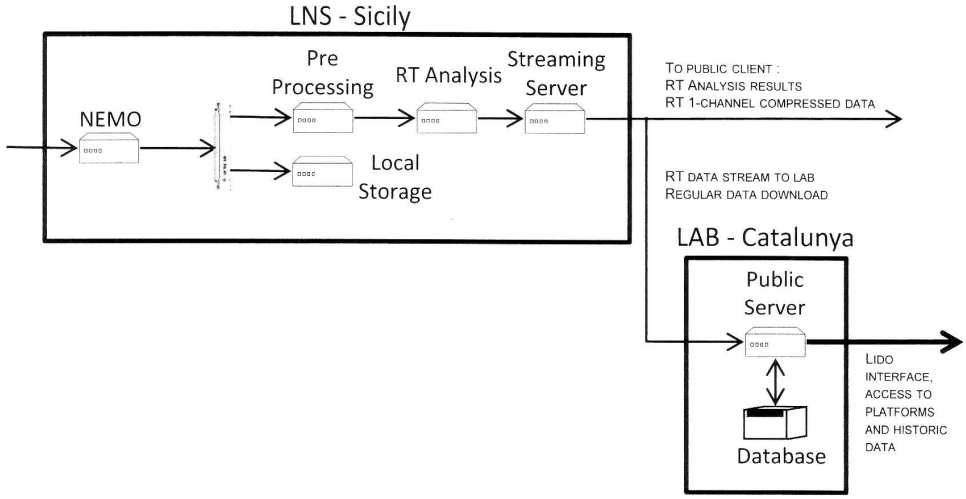

Figure 3.2: LIDO system design as it will be implemented for the NEMO site. The real-time processing is done with the help of two servers, the pre-processing server, designed to process data close to the source, performs detection of acoustic events, and the real-time analysis server handles source localisation and classification. The results are made available directly to the public from the platform site and are sent to the LAB for post-processing.

3.2 Thesis Outline and Objectives

The thesis will contain the following sections, which follow the work that has been carried out in the last five years. In Chapter 4 general background information will be presented on the current understanding of sperm whale acoustic signals, the production mechanism, their directivity properties, and on the applications of tracking sperm whales.

A first approach at recognition of a sperm whale group through the use of the rhythmic pattern in coda vocalisations is presented in Chapter 5. While this would not allow tracking of an individual animal during its dive, it is biologically interesting to automatically identify a specific social group and to track the group as a whole. This method was motivated by studies that have been done in the past that found different coda patterns for sperm whale groups (e.g. [73, 101]), although these studies do not usually define an objective method for discrimination. Classification was usually done very subjectively by the researchers, making it impossible to qualitatively compare results between different studies. An objective coda-labelling algorithm is proposed and results are presented on data from the Canary Islands. The results found did not justify a more in-depth analysis of coda based classification, and attention was moved towards click based identification.

In Chapter 6 an attempt at whale separation with the use of the Gabor function is described. Most of this work was done when there was still little knowledge concerning the directivity of a click, and under the assumption that a click could be characterised using only a few dominant frequencies from its wideband spectrum. The motivation of the use of a Gabor came from its successful application of modelling single dominant frequencies in dolphin sonar (although dolphin classification was never attempted). The approach performed poorly on sperm whales, which can be explained by the fact that clicks are significantly influenced by directivity. A more flexible method was needed, not restricted to a single frequency.

Chapter 7 explains the approach that was followed to extract characteristic information from clicks that could be used to distinguish between animals. The choice was made to construct a local discriminant basis using wavelet packets. This technique expresses a signal in a redundant basis, which then allows the selection of a particular basis that emphasises differences between classes. The advantage of this method was that, without trying out many different algorithms, it was still possible to select an optimal decomposition of the signal in time-frequency bands from many different choices. Classification was tried based on the selected coefficients using a linear algorithm, but this turned out to be insufficient.

Analysis of the coefficients that were selected from the wavelet-packets, which showed a tendency to cluster, led to the use of a radial basis function neural network for classification, as detailed in Chapter 8. This type of network can be split in two steps. First, the data (or available classes) is modelled using a clustering algorithm, and then the combination of the distances of a given pattern to the centres of the clusters is used to classify patterns. The advantage of this method was that after the clusters were modelled, classification of patterns was very fast, easily allowing real-time classification of data.

While the radial basis function network performed well, another type of non-linear classification was tried to see if the results could be improved. Section 8.4 explains the classification using support vector machines. This classifier is especially known for its capability to generalise to unknown data. Its structure can be described identical as RBF networks (a linear sum of non-linear mappings), but differs in its training approach where it focusses on the boundaries between two classes instead of class centres of multiple classes. As SVM are designed to solve the two class classification problem, a one-against-one strategy was used where the class with highest frequency was selected. For seven classes this led to 21 machines. Performance was found to be similar to RBF, but at the cost of a higher complexity and computation time.

The main problem with the classification algorithms described above is that they need to be trained.

It might not be possible to find values for time-frequency coefficients that are always useful for separation of individual animals, let alone values that could always be characteristic for an individual animal (which would have allowed the creation of a dictionary with acoustic features, as is done with photos for photo-identification). Not enough data is available to investigate the variability of features from the same animal in different recordings under different circumstances, thus for now the safest assumption would be to assume that the coefficients will vary. This means that, at the start of each recording, a number of clicks need to be used to train the algorithm which requires a way to separate the first few clicks. This algorithm can be allowed to use more time, as it runs only once and then passes the results on to the fast RBF classifier. This problem will be addressed in Chapter 9 where the use of a clustering algorithm based on a variational mixture of Gaussian distributions is discussed for obtaining the training data.

Finally, directions for continuing research and concluding remarks are made in Chapters 10 and 11. The data used to test the classifiers and coda analysis is described in detail in Appendix A will be explained in detail.

Objectives

The thesis objectives can be summarised as follows :

1. Investigate the use of codas for sperm whale group identification
2. Investigate the Gabor Function as a model for dominant clicks in the sperm whale signal
3. Find a method to extract discriminating features from sperm whale clicks
4. Find a classifier that allows distinguishing between sperm whales based on acoustic cues in real-time
5. Propose an unsupervised algorithm that can reliably create a training data set

Chapter 4

Background

4.1 Introduction

Most acoustic research on marine mammals has been done on the 'interesting' sounds, like baleen whale vocalisations and dolphin whistles. In the case of baleen whales, the use of sonar for echo location is rare (an interesting discussion about humpback sonar can be followed here: [36], [12], [61]) as they produce sounds in the infrasonic range, below 20 Hz. This makes their vocalisations audible at distances of more than a hundred kilometres away, and therefore interesting to study. Most interest is in categorising the different signals from the species, and tracking how signals of an individual can evolve over time (e.g. [74, 42]). Especially the humpback whale is famous for its whale songs and has been studied extensively, although little has been published on acoustic identification. In terms of species differentiation one study is for example [45] where wavelet features were used to differentiate between porpoises (exact species unspecified) and sperm whales. From a bio-acoustic point of view this article can be considered somewhat limited as it does not focus very well on the acoustic signals themselves, using species with completely different vocalisations and only sampling at 25 kHz; e.g. harbour porpoise sonar (which may not have been included in the data) has a peak frequency around 120 kHz [11] while sperm whale sonar has a peak around 12 kHz [65].

Similar research has been done on dolphin whistles, significant attention has been given to extraction of whistles from data, using time-domain techniques or image extraction methods from a time-frequency representation (e.g. [87, 88]). Additionally, attempts have been made to classify whistles in types that may be characteristic for an individual ('signature whistle'), a social group, or geographical location (e.g. [21, 60, 9]). The same kind of whistle-classification research has been done on orcas (e.g. [69, 92]). Using signature whistles for identification is still somewhat controversial, while dolphin sonar signals are generally not considered for identification purposes. However, there has been work in modelling dolphin sonar signals, mainly by attempting to fit a Gabor function [37] on the signal [48, 49]. The Gabor function is interesting to consider, as it exhibits an optimal time-frequency resolution (in fact, it is the only type of function with this property), and the ability to use it as a model on sonar signals would indicate an optimal use of their signal by dolphins. The same was tried on sperm whales, in an attempt to use its parameters for classification [97], but as will be described in Chapter 6, the study found that it was not as easy to fit the model on sperm whale clicks, likely due to click directivity reasons.

There has also been extensive research on bat sonar signals, including identification of different species. One example paper, that gives a good overview of previous bat related research, is [72], which compares the identification performance between discriminant function analysis and a perceptron neural network on twelve different bat species. As bats often seem to include a frequency modulated component into their

signals [11], the analysis of their sonar is best compared to dolphins, where similar techniques are used for especially whistle classification. A function that has been used to model the bat type of sonar is the hyperbolic frequency modulated (HFM) waveform [3, 32]. However, this kind of FM characteristic is rarely found in sperm whale sonar, which almost exclusively consists of short impulsive signals (Section 4.3) where all frequencies arrive instantaneously. Therefore, the Gabor function was preferred for modelling a single component of sperm whale sonar, instead of the HFM or other chirp-like models.

When looking at acoustic research on sperm whales, most interest has gone to coda classification (Section 5.4). Just as with dolphin whistles, this is concentrated on classifying coda types, finding different dialects between groups in different geographical locations, or on finding codas that are specific for a social group (e.g. [100, 104, 81, 94]). Additionally, at least one attempt has been made to separate and recognise individual animals based on their codas [46], this will be detailed in Section 5.2, together with the recognition techniques based on the sperm whale sonar click. Finally, in [31] classification is tried using time-frequency characteristics, but fails when the classifier is only trained with a small part of the data set; this is again due to the variability of the features during a dive.

The thesis work has been focussed on sperm whales, partly for the biological interest in their acoustic identification, which had not been successful so far, but also for very specific practical reasons. An important application is the tracking of sperm whales to avoid collisions in areas that contain heavy shipping, as explained in Section 3, but as important is the use of the whales as indicator of marine mammal presence in an area. Over the last few years the topic of noise pollution in the oceans and its effect on the environment has become a critical issue, for example in the construction and deployment of windmill farms or the planning of sea highways. Very little is known about the influence of background noise on cetaceans and assessing the effect of noise on marine mammals is difficult to measure, as their presence has to be detected in time periods before, during and after increased noise. Visual detection is costly and time intensive, while automatic acoustic based detection is practically very limited for most cetaceans. Most dolphin species use very high frequencies for their sonar (with components over 100 kHz), making detection difficult after a few hundred meters. Other signals like whistles, which are used by dolphins and for example orcas, are in a frequency band around 10 kHz making them audible over longer distances, but the characteristics of the signal itself, lacking a well defined peak, makes it more difficult to be used in simple localisation algorithms. Similarly, baleen whales use very low frequency signals that can easily travel dozens of kilometres, but at the same time this makes it difficult to position the sources unless a wide aperture hydrophone array is available.

Sperm whales, however, produce a sonar signal in a frequency range and at a level that allows it to be audible well over 5 kilometres in distance, while being short enough in time to be usable for localisation within the area around a small array. This makes them ideal candidates for detection, tracking and monitoring over long periods of time with limited equipment, which in turn will allow correlation between background noise levels and whale activity in a specific area. If such a relationship can be found for sperm whales, then this would provide the necessary stimulus to allocate funds for more costly research on the noise effects on other cetaceans. As such, the work developed for this thesis will be applied to monitoring systems that are currently developed within the ESONET framework, especially the LIDO project. LIDO will be implemented in underwater platforms in for example Italy and France (NEMO and ANTARES sites respectively), which both are situated in areas with sperm whale activity.

4.2 Sound Production by Sperm Whales

Little is known with certainty about how sperm whales produce sound, but a commonly accepted theory was suggested in [71] and refined throughout recent years in [66, 63, 64, 56]. Figure 4.1 shows a cross-section of a sperm whale head, which can take up to 30% of its weight and a quarter of its length [70]. The process starts

at the blowhole where air can enter through controlled breathing while the animal is at the surface. Most of the air passes through the trachea to the lungs, but part is moved through the right nare to fill the frontal air sac. To produce a click, the air is driven through the left nare and through the monkey lips by sphincter muscles. This produces a click, which is then reflected backwards because of the distal sac placed in front of the monkey lips. The click travels through the spermaceti and is then reflected a second time against the frontal sac which moves it forward again through the junk. This process of reflection and travelling through the spermaceti both amplifies and focuses the click, resulting in a source level of around 230 dB [65] when it leaves the front of the head. The production of clicks can be continued until there is no more air left in the frontal sac. It is assumed that a recycling process sends the air back from the distal to the frontal sac through the right nare passage.

Before the publication of [65], clicks were thought to contain multiple pulses, while there was some uncertainty about the degree of directionality. Older studies [13, 54, 98] found low source levels with almost no directivity characteristics, while other (mostly later) studies [106, 66, 91] did find a directional component. In [65] a small number of on-axis clicks were recorded which not only confirmed click directivity, but seemed to indicate a mono-pulse structure. Unfortunately, it is extremely rare to find these clicks on recordings. Otherwise, if it were possible to work with on-axis clicks, the classification task might be easier as the signal to noise ratio is much better and there is no strong influence from the other time-delayed multiple pulses. The multi-pulse structure as found on the off-axis clicks is probably caused by multiple reflections between the frontal and distal sacs, as illustrated in Figure 4.1 [107]. At every air sac reflection an echo is created that can be recorded by a hydrophone. These echoes are weak compared to the level of the main pulse (p1) when measured on-axis, but can appear strong in off-axis measurements. An example of an on- and off-axis recorded click and the directivity affect can be seen in Figure 4.2 [65], it should be emphasised that there is a 40 dB difference in the scales.

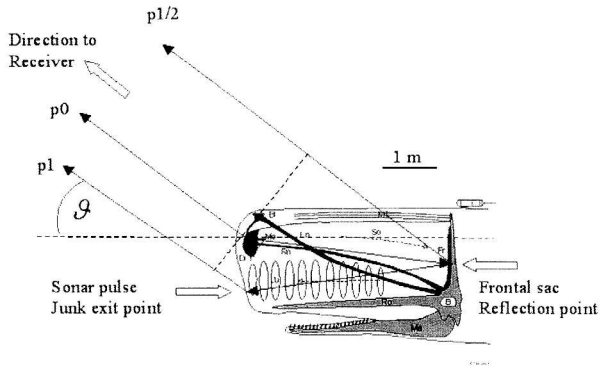

Figure 4.1: Cross section of sperm whale head, reproduced from [107]. **B** - brain; **Bl** - blow hole; **Di** - distal sac; **Fr** - frontal sac; **Ju** - junk; **Ln** - left nare; **Ma** - mandible; **Mo** - monkey lips; **MT** - muscle-tendon layer; **Ro** - rostrum; **Rn** - right nare; **So** - spermaceti organ. The image shows different pulses, apart from the main pulse p1, being produced by reflections against the air sacs.

4.3 Sperm Whale Vocalisations

Sperm whales generally produce only one type of sound, named a click. This is a broadband signal, containing frequencies from 200 Hz up to 32 kHz. The duration of a single click is often not very well defined. The maximum duration of a click is around 30 ms, while the first and strongest pulse usually takes less

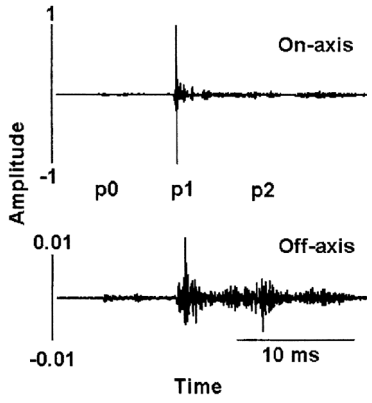

Figure 4.2: Example of an on-axis and off-axis recorded click, reproduced from [65]. The top image shows a click recorded on-axis with one strong pulse and low energy reverberations. The bottom image (with a different scale) shows the same click recorded off-axis; the first pulse is no longer as dominant over the reverberations.

than 5 ms. The variation of sperm whale sounds is mostly in the rhythm in which the clicks are produced. Several types of signals are recognised; first *normal clicks* are those that are produced during foraging, with an interval of around 1.2 Hz. These are generated in a series, called a click train, with occasional pauses or creaks (described below). This type of click is assumed to be used for echo location purposes. It has been estimated that this sonar is capable of detecting a 0.2 m squid at a distance of 1.2 to 2.2 km, depending on the sea state, and assuming reasonable directional hearing [6]. A school of squid would raise the detection range to several kilometres. The function of the pauses are not clear, it can be speculated to be used for air recycling, prey capture (although for example dolphins can continue their sonar while feeding), or in some cases it can be that the signal is simply too weak to be recorded.

The second type of signal that is recorded during foraging is called a *creak*. These are clicks with a high repetition rate, values have been reported from 50 Hz [13, 57] to 90 Hz [19, 99] to 220 Hz [40], with a total duration of around one minute. The function of a creak has been speculated to be a close range sonar [40], very similar to dolphin sonar when they approach their prey [62, 77].

The third type of signal that is recorded often, but only at the surface, are *codas*. This is a short sequence of clicks, three to twenty signals, that is repeated a few times in a certain rhythm. Codas are assumed to have a social role, although is it unclear what their exact function is. As mentioned in the introduction, there has been a large amount of research to find regional or group specific patterns, but their importance remains unclear.

One can also identify *rapid clicks* [99, 40] and *slow clicks* [101], which have repetition rates of around 15 Hz and 0.17 Hz respectively. These types are recorded less often; since they are used only at the surface, they are assumed to have social roles as well.

Recently a different type of sound has been reported, called *squeals* [102]. These sounds were perceived as tonal with a narrow band frequency-modulated structure, but actually consisted of clicks with a repetition rate of 713 to 1385 Hz. Tonal sounds have been heard before from sperm whales, but have not been described and are rare. The meaning and social context of the signals are completely unknown.

One question that can be answered now is why there is an interest in using sperm whale clicks. Although

highly directional, the clicks still contain enough energy to be recorded at large distances. Additionally, the frequencies contained in the signal are still high enough to be well-localised in time. For passive localisation purposes it is often necessary to match a signal on multiple hydrophones. These hydrophones can be together in a small array, or separated over a wide area. Especially in the first case a high precision matching process is needed to achieve accurate time delay of arrivals that are used to estimate the animal's location. For a small array, a 1 ms error can already make the positioning worthless. Another highly attractive feature is that sperm whales (at least while foraging) continuously produce sound. This means that a location estimate does not have to be based on a single click, but a large number of clicks can be used to produce a cloud of estimates on a map, and from this a position can be determined. A consequence of the high level and continuity of the clicks is that they may be used to detect other silent animals through a technique called Ambient Noise Imaging [79, 18]. This is still experimental but positive progress has been made [28]. A vital part in these applications is the ability to recognise an animal, at least within the group it dives with, in order to match the clicks recorded on different hydrophones.

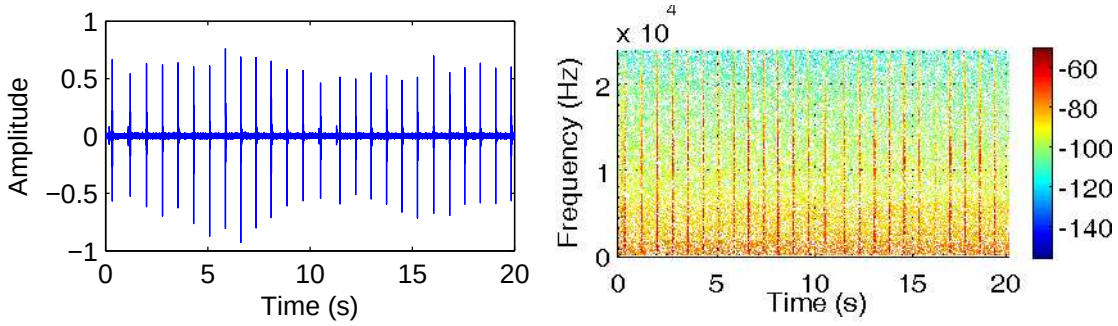

Figure 4.3: Regular clicks from a single animal. In this case the signal to noise ratio allows easy detection of the clicks, although surface echoes may be included too.

4.4 Detection of Sperm Whale Clicks

The first step in identifying a sperm whale is extracting its signal from the background noise. In Figure 4.3 a typical example of a segment of a recording is shown. In this case detection is quite easy and can be done with a simple threshold slightly above the noise level. This task can become much more difficult when the signal to noise ratio is poor. The broad band nature of a click makes it easily recognisable as a bar covering frequencies up to 20 kHz, as is shown in the right image of Figure 4.3, but this is not always the case. Higher frequencies may get lost more easily due to the distance to the animal and its orientation. There are several techniques that allow to discard some of the noise. Of course, the simplest is to run the recording through a high pass filter at around 400 Hz which remove much of the background sea noise. This method of detection where a conventional high pass filter is applied was used on the data in this thesis. A few other methods that have been proposed in the field are described below for completeness. They have been shown to perform well on specific data sets, but there is no commonly agreed on (or provably best) method for sperm whale click detection.

One proposed filter to reduce noise on a signal s is the following based on the Teager-Kaiser energy operator [50] :

$$S(n) = s^2(n) - s(n+1)s(n-1). \tag{4.1}$$

This filter will amplify high frequencies while attenuating low frequencies, making it easier to detect clicks in poor signal-noise conditions.

Another approach that has been suggested to filter out noise and emphasise a click in recordings is the use of the Hilbert-Huang transform [44, 1]. This transform decomposes the signal through a process called empirical mode decomposition into intrinsic mode functions (IMF), which are defined as functions where (1) the number of extrema and the number of zero-crossings are equal, or differ by one at most, and (2) the mean value of the envelopes defined by the local extrema is zero. The signal can then be written as

$$s(t) = \sum_{i=1}^{n} c_i + r_n, \tag{4.2}$$

where c_i are the modes and r_n is the residue. The modes are calculated by successively filtering the signal by the average of the local minima and maxima envelopes (m_i), i.e. $s(t) - m_1 = h_1$. The residue h_1 is not usually an IMF and it's used as the new input into the filter, $h_1 - m_2 = h_{11}$ until the result h_{1k} is an IMF, or a suitable stopping criterion has been reached. The final h_{1k} is then defined as the first mode c_1, and the same process is repeated on the residue $s(t) - c_1 = r_1$ until r_n becomes a monotonic function or the values become too small. An advantage of this method can be that the signal is decomposed onto itself, without using some family of orthogonal basis functions as with the Fourier or wavelet transforms. This makes the method in a way more independent. A time-frequency diagram can then be obtained by taking the Hilbert transform of (4.2) which allows the calculation of the instantaneous frequency. This decomposition can efficiently detect sperm whale clicks (or other non-noisy signals) because the low frequency noise will be mainly present in the higher modes of the decomposition, only requiring the first few to be calculated. Reconstruction of the signal will then permit easier detection of the clicks. Other approaches can be found in for example [68] or [55].

Once a segment of a recording has been recognised as a signal, and not noise, it needs to be confirmed that it constitutes a sperm whale click. Using frequency information it is not too difficult to confirm this. Clicks can be distinguished from, for example, dolphin whistles that can use a similar frequency bandwidth, by their duration and frequency signature. Clicks are short and form a spike in frequency, while whistles are longer and modulated. More thorough methods have been designed. For example in [38] the authors look at the energy distribution of the signal and take a measurement of the kurtosis to identify a cetacean signal.

4.5 Synchronisation of Sperm Whale Clicks

One difficulty for classification of sperm whale clicks can be the synchronisation of the clicks. When characteristic features are selected in the time-domain relative from a certain point t_0 defined as the start of the click, it is important that there is not too much variation in the estimation of t_0 for each following click. Figure 4.4 shows an example of 50 superposed low-pass filtered clicks from the same animal, where synchronisation was based on cross-correlation peaks. It can be seen that the energy distribution in the click changes gradually, likely due to a different time delay of a reverberation in the head as the angle between the direction of the call and the recording hydrophone changes.

A first conclusion that can be drawn from this example is that the reliability of the features that are extracted from the signal decreases for points further away from t_0. After 10 ms the signal becomes too noisy to be of use.

The second important point is that even for characteristics selected at the beginning of the signal a positional shift can be expected of a few samples between clicks. Working in the frequency domain with a signal decomposition by wavelets, this means that it is not likely that wideband features (which are narrow band

in the time domain) will be found. However, as will be explained in Chapter 7, the wavelet approach offers a wide variety of time-frequency bandwidth trade-offs and it should be able to cope with a small temporal shift in the features.

To give an impression of how the shape of a click changes during the dive, Figure 4.5 shows a superposition of three consecutive clicks (for frequencies below 3000 Hz) taken at the start, middle and end of a click train. There seems to be a pulse with changing phase delay, moving towards the left of the signal and creating a new peak around sample 200. This kind of behaviour will make it difficult to use techniques that purely rely on the signal's shape in the time domain.

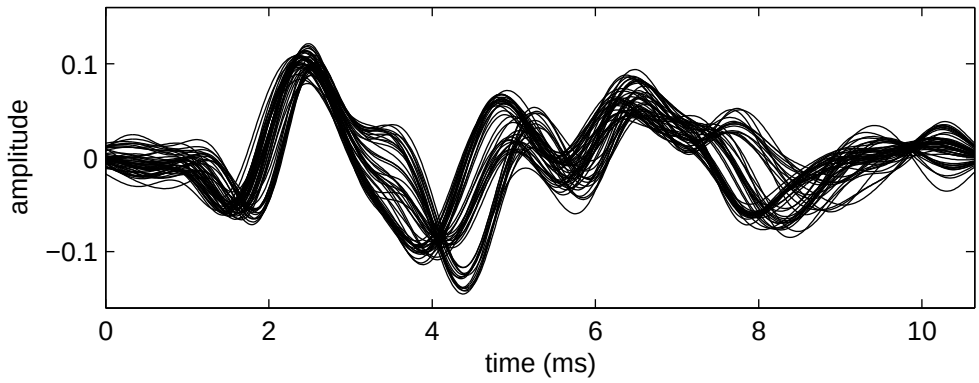

Figure 4.4: Superposition of 50 low-pass filtered clicks from the same animal. From this plot it should be expected that there may be a few samples error in the position of purely temporal features.

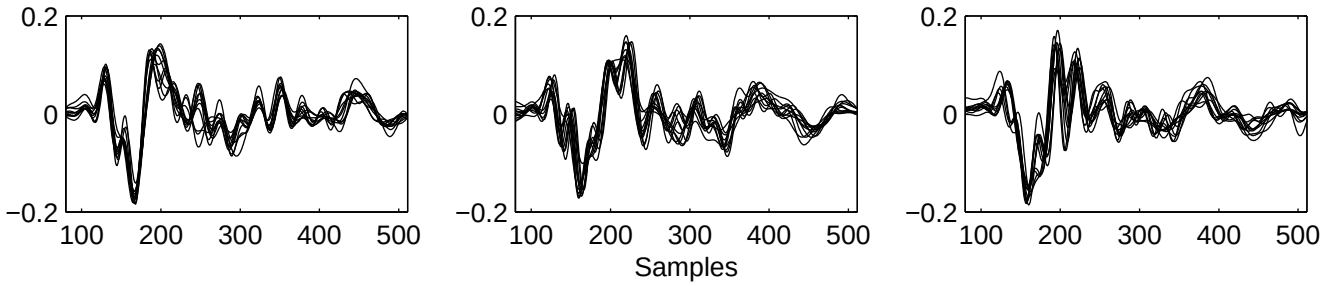

Figure 4.5: Evolution of a click (images show superposition of 10 subsequent clicks) in the course of a click train. A pulse can be seen moving from right to left, creating a new peak around sample 200.

Chapter 5

Clustering Rhythmic Patterns

5.1 Introduction

In the last ten years there has been an interest in the analysis of sperm whale codas, especially in their rhythm. As was explained in Section 4.3, these signals are usually produced in social situations and consist of a short sequence of clicks that is repeated several times. It has been speculated that codas can be specific for a sperm whale group and that there are geographically based differences in coda repertoires [105, 100, 104]. With the objective in mind of tracking sperm whales and possibly the social group they live in, it was interesting to see if this research in coda rhythm could be validated and extended to be used as an unique identifier. The clicks themselves that form the coda structure were not considered for identification as sonar clicks are much better candidates for this due to their much more regular occurrence. Since the coda data that was available for this study did not include any information on the individuals that produced them, the main aim of this analysis was to see if codas can be classified more robustly than is currently done. A robust and objective method will then serve as a tool for future research, to allow comparison of coda types between studies and investigate characteristics for individual animals.

The comparison of codas between different groups requires the codas to be classified into different types. This is normally done by normalising the codas by their duration to obtain the rhythmic structure (a different rhythmic structure may also be apparent in normal click trains [7]). Codas that contain the same number of clicks are then clustered based on their inter click intervals. In [81] the authors discuss two clustering methods based on k-means together with a classification-free approach that allow coda repertoires to be compared. It was concluded that there is a need for a robust method of classification and, for the data used, the criterion based on Duda-Hart's ratio performed best. However, this particular method does not seem to be robust with coda data that was recorded at the Canary Islands; based on visual observations at the time of the recordings the method showed a tendency to find an unreasonable large number of clusters. Most likely this was because the clusters did not follow the spherical model assumed in Duda-Hart's criterion. While perhaps a specific coda type can be expected to have a spherical distribution, a high number of samples may be required to obtain this distribution.

Unfortunately, a typical problem with codas is that the longer codas containing many clicks are less commonly found in recorded data. This can be especially problematic considering that higher dimensional data should have a larger number of samples for reliable density estimation. In cases where few coda samples of a specific length have been found, their analysis requires a method that has less restrictions on the model. Therefore it was decided to investigate the use of g-means [43] for coda classification. The g-means algorithm also relies on k-means to perform clustering, but it accepts clusters when they follow a normal distribution

in the direction of their principal component. The dimension reduction before the acceptance test should reduce the amount of samples needed for long codas.

In order to make sure that the classification is non-random and reproducible, the initialisation step of k-means was changed to be deterministic. Otherwise, it was found that the clustering process needed to be repeated over fifty times to ensure a stable result. This dependency of the clustering outcome on the initialisation makes it more difficult to compare results between different studies which requires a more consistent method. One way to obtain this is by initialising the centres by selecting samples that lie as far away from each other as possible [51]. With this approach the centres are likely to end up in different clusters and it makes the clustering algorithm run fast in a reproducible way.

In the following, the clustering algorithms that were used on codas are first briefly explained in Section 5.2. Then an alternative method of coda labelling is presented in Section 5.3 that allows improved comparison between data sets. The results of clustering the Canary Islands coda data set are presented in Section 5.4 and finally the outcome is discussed in Section 5.6.

5.2 Coda clustering

5.2.1 Duda and Hart's ratio criterion

First a brief explanation of one of the clustering algorithms that is commonly used in the field, the Duda and Hart's ratio criterion [33], is presented. The underlying assumption of the method is that the data can be described by normally distributed spheres, i.e. clusters containing constant diagonal covariance matrices. Defining the null hypothesis that data is described by a sphere, the null hypothesis is rejected if splitting the data improves the clustering error sufficiently so that the improvement cannot be explained by pure chance. The errors for the one cluster ($J_e(1)$) and two cluster model ($J_e(2)$) are given by :

$$J_e(1) = \sum_{x \in \mathcal{D}} \|x - m\|^2 \tag{5.1}$$

$$J_e(2) = \sum_{i=1}^{2} \sum_{x \in \mathcal{D}_i} \|x - m_i\|^2 \tag{5.2}$$

where $\mathcal{D}$, $\mathcal{D}_i$ are the original and split clusters and m, m_i their sample means. The null hypothesis can then be rejected at the p-percent significance level if

$$\frac{J_e(2)}{J_e(1)} < 1 - \frac{2}{\pi d} - \alpha \sqrt{\frac{2(1 - 8/\pi^2 d)}{nd}} \tag{5.3}$$

where d is the dimension of the data and α is defined by

$$p = 100 \int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du. \tag{5.4}$$

The algorithm continues splitting clusters with k-means as long as the ratio given in (5.3) keeps improving at the specified significance level.

5.2.2 Limitation of the spherical model

An important assumption in the Duda-Hart criterion is that the data are spherically distributed and this may not always be the case with coda data. As is shown in Table 5.2 in the result section, the criterion

seemed to find an unreasonable large number of clusters in the coda data from the Canary Islands based on visual observation of the number of animals in the area during the recordings. Normally, only animals close to each other and at or near the surface will produce codas, allowing an estimate of their number. Of course, not all whales were visible and it was generally not known which ones were acoustically active.

To see why the criterion overestimates the number of clusters it can be illustrative to look at its performance on data that has a smaller dimension than the number of observations. For this, test data was created consisting of two 3-dimensional clearly separated clusters, each containing 1000 samples. In the first case the dimensions of the first cluster were drawn from a N(0,1) distribution, whereas the dimensions of the second cluster were drawn from N(10,1). Then, to simulate the situation with coda data that have a lower intrinsic dimension (this is always the case when the codas are normalised), the two clusters were made flat by sampling the third dimension from N(0,0.5) and N(10,0.5) distributions. A thousand different data sets were created this way and subsequently clustered, taking the average of the number of clusters found. Each clustering itself was repeated 50 times to find an optimal and stable outcome.

The clustering results for the two cases are shown in Table 5.1. The first three rows show the variance of the two clusters in three dimensions as explained by their principal components. It can be clearly seen that in the first case the data is almost spherical while in the second case the data could be considered two dimensional. The next two rows show the number of clusters found by the Duda-Hart criterion, averaged over 1000 runs, using 95% and 99% significance levels. As can be expected, the criterion had no problem finding the two clusters when they were spherically shaped, but when they were flat the criterion had difficulties and needed to continue splitting the data until the clusters contained very few points so they resembled the model. The bottom row shows the result using g-means explained below, which does use dimension reduction and could find the two clusters in this case.

	Case I		Case II	
	cluster 1	cluster 2	cluster 1	cluster 2
pc 1 (%)	39	41	58	60
pc 2 (%)	33	34	42	39
pc 3 (%)	28	26	0.53	0.56
# clusters DH (0.95)	2		135.5	
# clusters DH (0.99)	2		66.5	
# clusters GM (0.9)	2		2	

Table 5.1: Evaluation of the performance of the Duda-Hart criterion in the case of spherical data (Case I) and flattened data (Case II). The first three rows give the variance of the clusters as explained by their principal components, showing the clusters to be flat in Case II. The next two rows give the number of clusters as found by the Duda-Hart criterion at 95% and 99% significance levels. The last row gives the clusters found by g-means. All values are averaged over 1000 runs.

5.2.3 g-Means

To make the clustering algorithm more flexible and less dependent on the model (and the intrinsic dimension of the data) a clustering method published as g-means [43] was investigated. This method also uses k-means to cluster the data, but uses a different model to accept clusters. Instead of using a high dimensional sphere, g-means projects the clusters found by k-means onto their principal component and then performs a 1-dimensional test for normality. If a group passes this test it is accepted as a cluster. The advantage is that the reduction of the dimension also reduces the complexity of the model allowing spherical but also flat

and elliptical normal distributions to be found. The test for normality is done with the Anderson-Darling statistic [4, 86, 83] described by

$$A^2 = \sum_{i=1}^{n} \frac{(1-2i)}{n}[lnz_i + ln(1 - z_{n+1-i})] - n \tag{5.5}$$

where $z_i = F(x_i)$ with F the cumulative distribution function of the $N(0,1)$ normal distribution and x_i the standardised data in ascending order. The usage of the sample mean and variance require the modified statistic [25] :

$$Z = A^2(1 + \frac{0.75}{n} + \frac{2.25}{n^2}). \tag{5.6}$$

Suitable critical values will have to be looked up from a table since the p-values are not easy to evaluate. In this report we used a critical value of 0.9 which relates to an α value of around 95%.

After every round with k-means, the clusters that pass the normality test are removed and the remainder of the data is clustered again with an updated k ($k_{new} = k_{old}-$ removed_clusters $+1$). A difference from [43] is that not all the clusters that fail the test are automatically split, instead the k-means algorithm is rerun with one additional centre. This is to prevent a situation where three clusters are divided over 2 centres, causing the normality tests to fail. Splitting both centres would lead to too many classes, while reclustering the complete data set with 1 extra centre allows the 3 classes to be found.

It should be noted that the normalisation of the codas excludes the possibility of clusters lying parallel to each other in the inter click interval space. This ensures that in most cases where the value of k in k-means is too low and multiple clusters are still grouped together, the projection of the clusters onto one dimension will result in a multi-modal distribution which will be rejected by g-means.

5.2.4 Initialisation of k-means

A common way to initialise the centres for k-means clustering is to select k points randomly from the dataset. The outcome of the algorithm then depends on this primary selection and one has to use multiple runs to find the most stable solution. Figure 5.1 shows two different outcomes of k-means classification with different initial centres. The clustering on the left converged to a local optimum while on the right it converged to the global optimum. In this particular case using a few runs will show the solution on the right to be more stable, but in general when the clusters are not as clearly identifiable as in Figure 5.1 the number of required runs can be large. In order to prevent having an outcome that relies on the number of runs it was decided to initialise the centres in a straightforward manner. For the first run, where k is 2, one centre is calculated by taking the mean of the data and the second centre is put at the point furthest from the first, based on [51]. For consecutive runs with an increased k the centres of the previously found clusters are reused, adding an additional centre to split up one cluster. This additional centre is placed within the cluster as far as possible from its centre. The initialisation does not prevent locally optimal solutions but generally gives good results.

5.2.5 Limitations of the clustering method

The k-means algorithm can still behave poorly when the clusters are not spherically distributed or insufficiently separated. In that case the normality criterion can overestimate the number of classes by a large value. This is illustrated in Figure 5.2 where two elliptically shaped clusters are used as input. Both are normally distributed on the x and y axes with different variances. The first image on the left shows an initial clustering by k-means. Due to the short distance and elongated shapes, the optimal centres for k-means mix

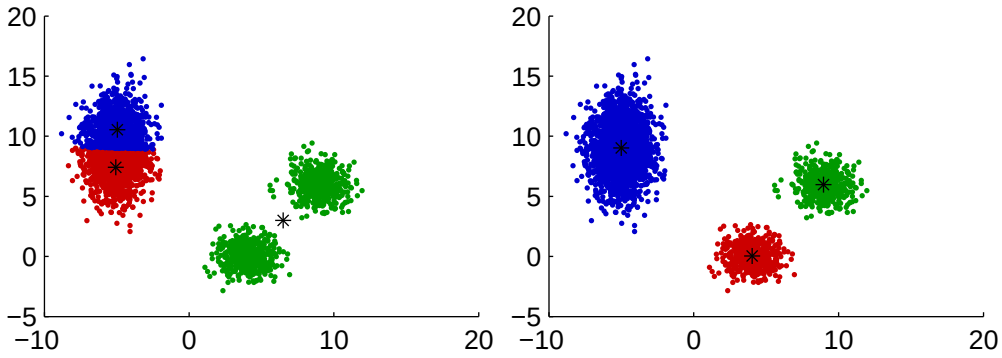

Figure 5.1: Local and global optimal solutions for the k-means algorithm based on different initial centres.

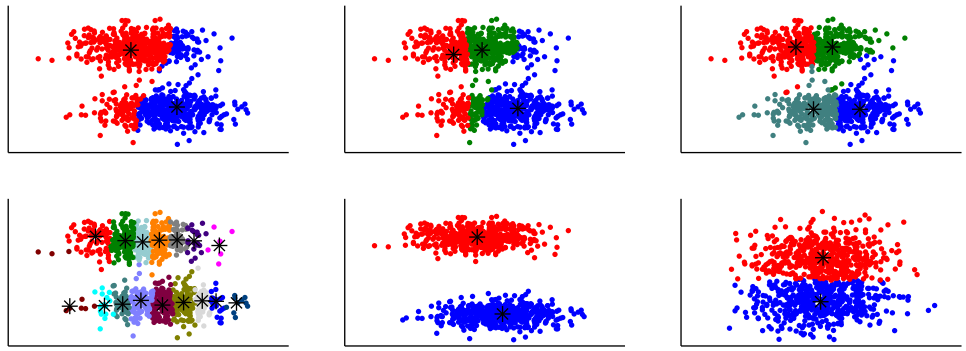

Figure 5.2: From left to right the first four images show the overestimation of the number of classes when the data is ellipsoidal and dense. The last two images show the improvement when either the data is sufficiently separated or has a more spherical distribution.

the two clusters. This ensures that the test for normality fails and a centre is added, which is displayed in the second image of Figure 5.2. With 3 centres the two clusters are still mixed and the normality tests fail again. The next step using 4 centres is able to separate the clusters, however the tests for normality will fail because none of the 4 subclusters have a normal distribution in the direction of their principal components. It requires many more centres before the algorithm can finally stop, shown in the fourth image.

When the number of clusters are overestimated in this way, it could be tried to merge classes that are 'close' to each other allowing to detect more arbitrarily shaped clusters, but this complicates the protocol and should be unnecessary in most cases. Assuming that codas can be classified by their rhythm, it should be expected that the clusters are distributed around their centres with small variance and sufficiently separated to be detectable. In these cases the method will perform well, as shown in the last two images of Figure 5.2. The fifth image displays the two ellipsoidal clusters with sufficient distance to be classified correctly. The sixth shows two more spherically shaped clusters at close distance which can also be separated.

5.3 Coda Labelling

After the clustering process, current naming schemes focus on counting the number of clicks within a coda, grouping together those clicks that are 'closer' together. The definition of 'closer' is left to the researcher

and can be arbitrary. An example is given in Figure 5.3, which shows two normalised 5-click codas (the vertical bars are the coda's clicks). The coda on the left would be called '1+2+2' which could also be the label for the coda on the right, discarding information about the different time intervals. In the event that the codas were found in different studies, comparison of the coda types '1+2+2' would lead to erroneous conclusions. If both coda types were found in the same study, then the left coda may be renamed to for example '1+2++2' to indicate the different rhythms, but without any other information about the meaning of double or more plusses, comparison between studies is impossible. Authors need a certain amount of creativity and imagination to label the clusters found, using a mixture of characters and plus signs.

Figure 5.3: Both 5-click codas might be named '1+2+2' if they were encountered in different studies, while they follow different rhythms. When found in the same study the names might be '1+2++2' and '1+2+2' still giving little information about the rhythms and how they can be compared.

The proposed protocol aims to avoid any kind of subjectivity in the labelling of coda groups. As with the current naming schemes there are two parts that make the label, the characterisation of inter click intervals and the grouping of consecutive clicks that have similar lengths.

Click interval labelling Starting with labelling the intervals, it is assumed that the codas are normalised by duration, which is the case in practically every coda study. The normalisation implies that codas are analysed by their rhythm and suggests the use of a labelling algorithm that follows this rhythm, the inter click intervals, closely. A systematic approach requires a partitioning of the click interval lengths in a number of types. Current schemes make use of a regular interval type, where the clicks are more or less evenly spaced, and a number of plusses that signify longer intervals relative to the shorter intervals in the specific coda cluster.

Putting this in a more structured and objective frame-work, a perfectly regular coda consisting of n clicks would have click intervals of $\frac{1}{n-1}$. Using this interval as a unit length, a *very short* (VS) interval can be defined as an inter-click interval shorter than one third of the unit length, $\frac{1}{3(n-1)}$. Likewise a *short* (S) interval can be defined as shorter than two thirds of the unit length, or $\frac{2}{3(n-1)}$ and a *regular* (R) interval as shorter than four thirds of the unit length, $\frac{4}{3(n-1)}$. Figure 5.4 gives a clear overview of the labelling of the click-intervals. The remaining area between regular intervals and the maximum interval length can be divided in two to give *long* (L) and *very long* (VL) click intervals. Using this scheme the name for the left coda in Figure 5.3 would be labelled *'regular short long short'*, or 'R+S+L+S'. The coda on the right would be labelled *'long short long short'* or 'L+S+L+S'. Instead of counting clicks both labels describe the inter click intervals and immediately give an idea of the coda groups' rhythms.

Combining click intervals The other part of labelling is the combination of clicks. Obviously, when two consecutive click intervals have the same duration and carry the same label, they are combined, as for example *'two-long two-short'* or '2L+2S'. However it is possible that two intervals fall just on either side of a label boundary, as demonstrated in Figure 5.5 for a 4-click coda. On the left of the figure the coda is shown with 2 fast click intervals and 1 very long. On the right of the figure the length of the click intervals of this specific coda are drawn as the dashed vertical bars and the shorter vertical bars mark the label boundaries.

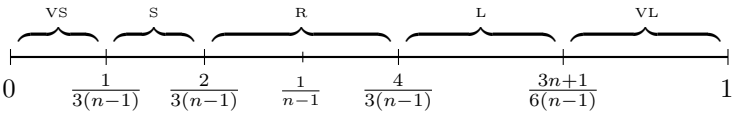

Figure 5.4: Classification of a normalised click interval of an n-click coda. The line represents the maximum length of an interval and is divided in 5 segments (*very short*, *short*, *regular*, *long* and *very long*) which depend on n. For example an interval of 0.15s of a 4-click coda would fall between $\frac{1}{9}$ and $\frac{2}{9}$ and be classified as a *short* interval.

While there is no problem labelling the coda as 'VS+S+VL' it can be preferred to group click intervals when they have similar lengths and to label their average duration (as is done with current labelling schemes). Since this protocol aims at removal of subjectivity this needs to be clearly defined. One straight-forward method is to use information from the cluster itself and to take into account the standard deviations of the intervals. Then two click intervals are considered to have similar duration if their means lie within 2 standard deviations of each other, or

$$\frac{|m_1 - m_2|}{s_1 + s_2} < 1 \tag{5.7}$$

where m_1 and m_2 are the sample means and s_1, s_2 the sample standard deviations of the intervals. If a third click interval m_3 following m_2 with standard deviation s_3 is also closer than $s_2 + s_3$ than the three clicks are combined. After the intervals are added together their new combined mean is evaluated and they are labelled together.

The labelling process is independent of the clustering algorithm and it cannot be guaranteed that different clusters receive different labels. The quality of any clustering algorithm highly depends on the data, and when the data separate poorly or do not conform to the model used for clustering, the cluster distances may be very small leading to identical names. Depending on the variances of the clusters it could be preferred to combine them in that case, but when necessary, the resolution of the labelling protocol can be improved by partitioning it into smaller intervals and dividing the unit length by a factor higher than three. However, again the standard deviation of the click intervals should be taken into account. For example the *short* intervals from the 4-click codas in Table 5.3 have standard deviations of 0.043, 0.059, 0.093 and 0.098 while the 4-click *short* interval spans a length of 0.11. Already the standard deviation of the last classes indicate that many click intervals fall outside of the *short* group. When the partitioning is further refined the characterisation of an average value will often no longer represent the majority of the intervals.

Figure 5.5: An example of a 4-click coda, on the left the normalised coda with four clicks and three numbered intervals, on the right the click interval lengths (three dashed vertical bars) plotted inside their labelling intervals bounded by the smaller vertical bars. Depending on the cluster's spreading it would be preferred to combine the click intervals 1 and 2 and characterise them together.

5.4 Coda Classification results

The clustering algorithm was used on the normalised codas found in recordings made at the Canary islands. The critical value used for the Anderson-Darling test was 0.9 for all coda lengths. The 3-click codas are

discussed first as they are somewhat more difficult to classify than the other types.

Normalisation of a 3-click coda with only two intervals leads to a representation that has only one independent dimension. In Figure 5.6 the top line plots the first interval of all the 3-click codas in the Canary data sets. Especially the slower codas seem to follow a uniform rather than normal distribution. The data only shows two obvious clusters both with large variances. Further subclustering to obtain classes with smaller variances can be done arbitrarily since there are no other obvious class boundaries visible. Using the Gaussian criterion with a critical value of 0.9 18 clusters were found; these are displayed in the second line of Figure 5.6. As can be expected the algorithm performs poorly on the slow intervals. Adjusting the critical value gives some control on the maximum number of clusters. The following lines in Figure 5.6 show the clustering for critical values of 1.3, 1.9 and 2.3 respectively. In this case the value 1.9 might be acceptable, but considering that the data can be clustered arbitrarily it is preferred to use the naming scheme to limit the number of classes. Combining identically named clusters only allows 5 possible classes for a normalised 3-click coda.

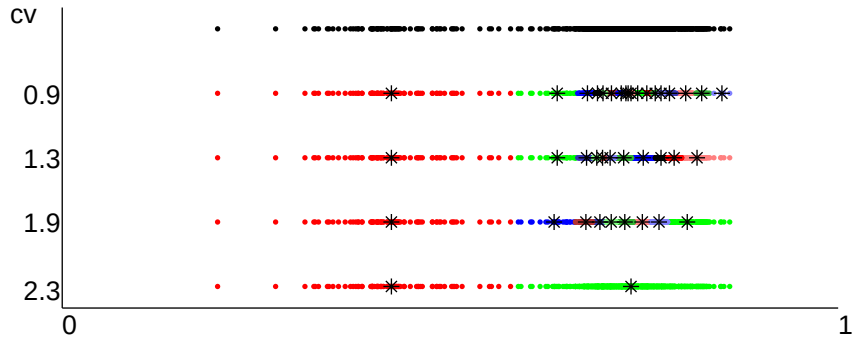

Figure 5.6: The top line shows the first interval of the normalised 3-click codas, with almost uniform distribution for slow intervals. The following lines show the result of clustering with increasing critical values.

To evaluate the performance of the modified g-means algorithm the coda data was also clustered using two algorithms proposed in [81], the results are shown in Table 5.2. The output of the variance-ratio algorithm is just given for reference; the algorithm itself has not been further discussed here as it was not considered a 'best choice' by the authors in [81] and it requires a human operator to select the number of clusters based on a specific criterion. In some obvious cases this could be automated but this was not researched further. For the Duda-Hart criterion two different significance levels were tested. The 95% value is normally used but to force a lower number of clusters found by the algorithm this was increased to 99% (an arbitrary choice). The g-means approach was tested with two different initialisation routines (random initialisation combined with repetition of the clustering process to find a stable solution and fixed initialisation as discussed in 5.4) to see if there were large differences in the outcomes. One conclusion that can clearly be drawn from the table is that clustering of the coda data is not obvious in the sense that there is an easily identifiable optimal clustering. The algorithms rarely seem to agree on the number of clusters in the data. This indicates how difficult it may be to find well defined coda-types in general, and also how difficult it will be to compare outcomes between different studies. It is vital that any published clustering result is accompanied by means and variances of the clusters. Otherwise, it is difficult to objectively conclude that one algorithm performs more reliable than another one; based on visual observations at the time of the recordings, g-means gave reasonable and stable outcomes for all coda types where Duda-Hart had some large over-estimations even using 99% significance.

coda length	VR	DH 95%	DH 99%	GM (random)	GM (fixed)	# codas
3-click	3	183	86	18	18	859
4-click	6	28	8	4	4	170
5-click	2	8	2	3	3	117
6-click	2	37	12	9	5	325
7-click	3	11	5	4	7	71
8-click	2	4	2	6	6	24
9-click	3	5	5	4	4	24
10-click	2	3	3	1	1	21

Table 5.2: The number of clusters found in the Canary Islands codas using three different clustering approaches. The variance ratio criterion tends to underestimate the number of clusters while the Duda-Hart criterion accepts a too large number of clusters at both 95% and 99% significance levels. The results with the adapted g-means algorithm seem reasonable without much difference between random and fixed initialisation. The 3-click codas form a special case, their distribution does not follow a normal distribution causing both DH and GM to fail.

The complete clustering results are shown in Table 5.3. Codas with more than 10 pulses were extremely rare and are not added to the table. All identified clusters are shown, including those with very few members. Although these cannot be considered a class, when the results are compared to other data they might appear more frequently. The Z value is the modified Anderson-Darling statistic, this value was not evaluated for very small classes. It is noted that for the 3-pulse coda the values were derived from combining all clusters with identical names, while the statistic is the average of the statistics of the individual clusters. For none of the other coda types identical cluster names from the naming protocol were encountered. It is remarkable that in 859 3-pulse codas there was not a single one that starts with a fast interval. Otherwise very fast or very short intervals seemed to be rare and these codas can be considered outliers. It is suggested that future studies present similar tables that will provide a complete overview of the data, objective naming of coda types, and will allow direct comparison of results.

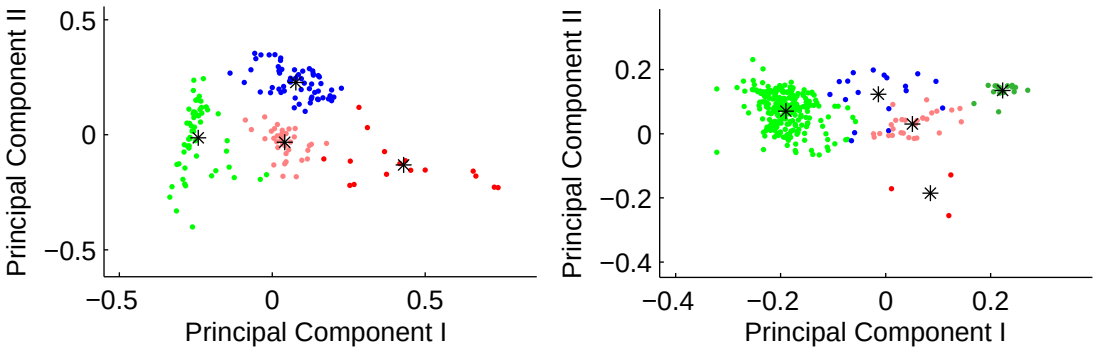

Figure 5.7: Clustering results for 4-click (left) and 6-click (right) codas, plotted against the data's two principal components. Any visible overlap between the clusters is likely caused by the reduction of the dimensions.

Table 5.3: Coda rhythm classification

class name	interval means $*10^{-1}$(std $*10^{-2}$)	#	Z
3-click codas		859	

Table 5.3: continued on next page

Table 5.3: continued

class name	interval means $*10^{-1}$(std $*10^{-2}$)					#	Z
2R	4.8(11)	5.2(11)				105	0.85
1L + 1S	7.4(4.0)	2.6(4.0)				749	0.64
1VL + 1VS	8.5(0.66)	1.5(0.66)				5	-
4-click codas						170	
3R	3.0(5.3)	3.3(3.5)	3.7(4.8)			38	0.45
2S + 1L	1.6(9.3)	1.4(9.8)	7.0(15)			16	0.51
2R + 1S	3.6(12)	4.9(8.8)	1.5(5.9)			57	0.75
1L + 1S + 1R	5.0(5.9)	1.6(4.3)	3.4(7.2)			59	0.47
5-click codas						117	
4R	2.3(3.2)	2.4(6.7)	2.4(6.1)	2.9(7.5)		37	0.73
1L + 3R	4.0(6.0)	2.0(6.6)	1.8(6.2)	2.2(7.4)		53	0.72
1S + 1R + 2L	0.97(3.8)	2.4(4.8)	3.1(6.3)	3.6(5.7)		27	0.52
6-click codas						325	
5R	1.7(3.2)	1.7(2.9)	1.8(2.7)	2.0(3.7)	2.7(6.1)	29	0.30
1VS + 1L + 3R	0.65(6.0)	3.6(5.7)	1.8(10)	1.8(6.4)	2.2(6.0)	3	-
2VS + 2R + 1L	0.65(1.8)	0.58(2.1)	2.3(3.2)	3.0(3.3)	3.5(3.2)	17	0.71
1L + 1R + 2S + 1R	3.8(4.1)	1.7(4.8)	1.1(2.6)	1.3(2.6)	2.1(3.2)	258	0.80
1R + 2S + 1R + 1L	2.6(5.6)	1.1(6.1)	0.96(3.5)	1.9(6.5)	3.4(7.4)	18	0.63
7-click codas						71	
5R + 1L	1.2(4.1) 3.0(4.1)	1.1(2.8)	1.3(2.1)	1.5(2.4)	1.8(3.2)	13	0.74
1L + 2S + 3R	3.5(4.6) 2.1(6.3)	1.2(4.6)	0.90(1.7)	1.0(2.0)	1.3(2.3)	41	0.77
1S + 1L + 4R	0.57(2.2) 1.9(2.8)	3.5(2.1)	1.6(1.0)	1.2(2.9)	1.1(1.6)	3	-
2VS + 1VL + 3VS	0.35(3.1) 0.12(1.2)	0.13(1.4)	9.0(9.4)	0.035(0.43)	0.36(3.2)	2	-
1R + 3S + 1R + 1L	2.0(4.9) 3.6(2.1)	0.51(0.46)	0.66(0.70)	1.0(3.0)	2.2(3.3)	4	-
1S + 3VS + 1S + 1VL	0.97(2.4) 6.5(2.1)	0.52(2.6)	0.41(4.1)	0.54(4.1)	1.1(8.2)	2	-
1L + 1S + 1L + 1S + 2R	2.4(3.2) 1.6(5.2)	1.1(4.6)	2.3(6.5)	0.62(4.0)	2.0(9.2)	6	0.25
8-click codas						24	
7R	1.4(2.4) 1.4(1.4)	1.4(1.5) 1.5(3.1)	1.4(1.2)	1.4(0.81)	1.4(1.1)	9	0.49
1L + 4S + 2R	3.7(5.9) 1.1(2.5)	0.92(2.2) 1.8(7.5)	0.82(1.3)	0.84(1.2)	0.77(1.7)	5	-
1S + 1L + 5R	0.57(7.4) 1.6(5.3)	2.6(5.1) 2.0(4.0)	0.92(3.4)	1.1(2.5)	1.2(3.0)	5	-
1R + 2S + 3R + 1L	1.2(6.2) 1.6(2.7)	0.86(0.55) 3.7(3.6)	0.68(2.1)	0.92(4.0)	1.1(1.3)	3	-
2R + 1L + 3R + 1L	1.0(-) 1.0(-)	1.2(-) 1.9(-)	2.7(-)	1.3(-)	0.91(-)	1	-
1L + 1R + 2S +	2.1(-)	1.0(-)	0.60(-)	0.78(-)	1.6(-)	1	-

Table 5.3: continued on next page

Table 5.3: continued

class name	interval means $*10^{-1}$(std $*10^{-2}$)					#	Z
1R + 1L + 1R	2.8(-)	1.1(-)					
9-click codas						24	
8R	1.2(3.5)	1.2(2.0)	1.1(3.0)	1.2(2.3)	1.1(1.8)	14	0.55
	1.2(0.78)	1.3(1.8)	1.7(5.1)				
2L + 3S + 3R	3.2(2.8)	1.8(1.7)	0.68(1.8)	0.66(1.0)	0.63(0.61)	7	0.36
	0.77(1.2)	0.98(1.5)	1.4(2.5)				
5VS + 1VL + 2S	0.40(3.0)	0.11(0.85)	0.11(0.83)	0.14(0.87)	0.22(1.4)	2	-
	7.9(17)	0.50(4.5)	0.58(5.2)				
1L + 1R + 2S + 1R +	2.4(-)	1.0(-)	0.55(-)	0.76(-)	1.5(-)	1	-
1VS + 1L + 1R	0.40(-)	2.4(-)	1.0(-)				
10-click codas						21	
8R + 1L	0.90(2.4)	1.1(5.3)	1.2(7.3)	0.95(1.7)	0.95(1.6)	9	0.54
	1.0(1.9)	1.0(2.0)	1.1(2.6)	1.7(5.7)			
1L + 1R + 3S +	3.0(4.3)	1.1(4.2)	0.60(1.1)	0.64(0.88)	0.65(0.88)	9	0.60
3R + 1L	1.2(-)	1.4(-)	1.8(-)	1.5(3.2)			
1L + 3S + 1R +	2.1(0)	0.68(0)	0.60(0.95)	0.58(0.95)	1.1(0.95)	2	-
1L + 2R + 1VS	1.6(0.95)	1.5(0)	1.4(0)	0.36(0)			
1L + 1R + 3S + 1VS +	2.8(-)	1.0(-)	0.74(-)	0.65(-)	0.69(-)	1	-
1R + 1L + 1S	0.23(-)	0.86(-)	2.4(-)	0.58(-)			

Table 5.3: Classification of normalised codas with standardized names. Z is the modified Anderson-Darling statistic and was not evaluated for very small classes.

5.5 Standardisation of Coda Analysis

This chapter has argued a few times the importance of a standard method for coda analysis and classification. To facilitate this, a tool (available from [26], Figure 5.8) was developed in MATLAB allowing researchers to apply the coda clustering and labelling methods. In addition to the algorithm described here, the tool includes methods developed by Luke Rendell and Hal Whitehead [81] to allow their performances to be compared together and the user can then choose a clustering approach that works best on their available data. Included in the analysis results are the variances for each click interval so the validity of the labelling can be assessed directly. It is expected that the use of a standardised approach will allow to first test the validity of rhythmic coda clustering techniques and afterwards to test classification of sperm whale groups or individuals based on a coda's characteristics, once enough data has become available.

5.6 Conclusion

The coda clustering method that was presented in this chapter performed well on the data recordings from the Canary Islands. The over-estimation of classes that can often be seen with the current existing methods seemed to be reduced especially on small data sets. Of course, there is no generally accepted existing method and as such it is always difficult to compare results between studies. There are two aspects in specific that can be criticised in current publications, first the objectivity when a label is chosen for a coda type. Without any kind of formal specification it is likely that different researchers will pick different names for their classes, and there is no easy way to link the coda types between studies. Second, not all publications pay attention to the variances found in the inter click intervals of the coda types for labelling purposes (e.g. in [85] or

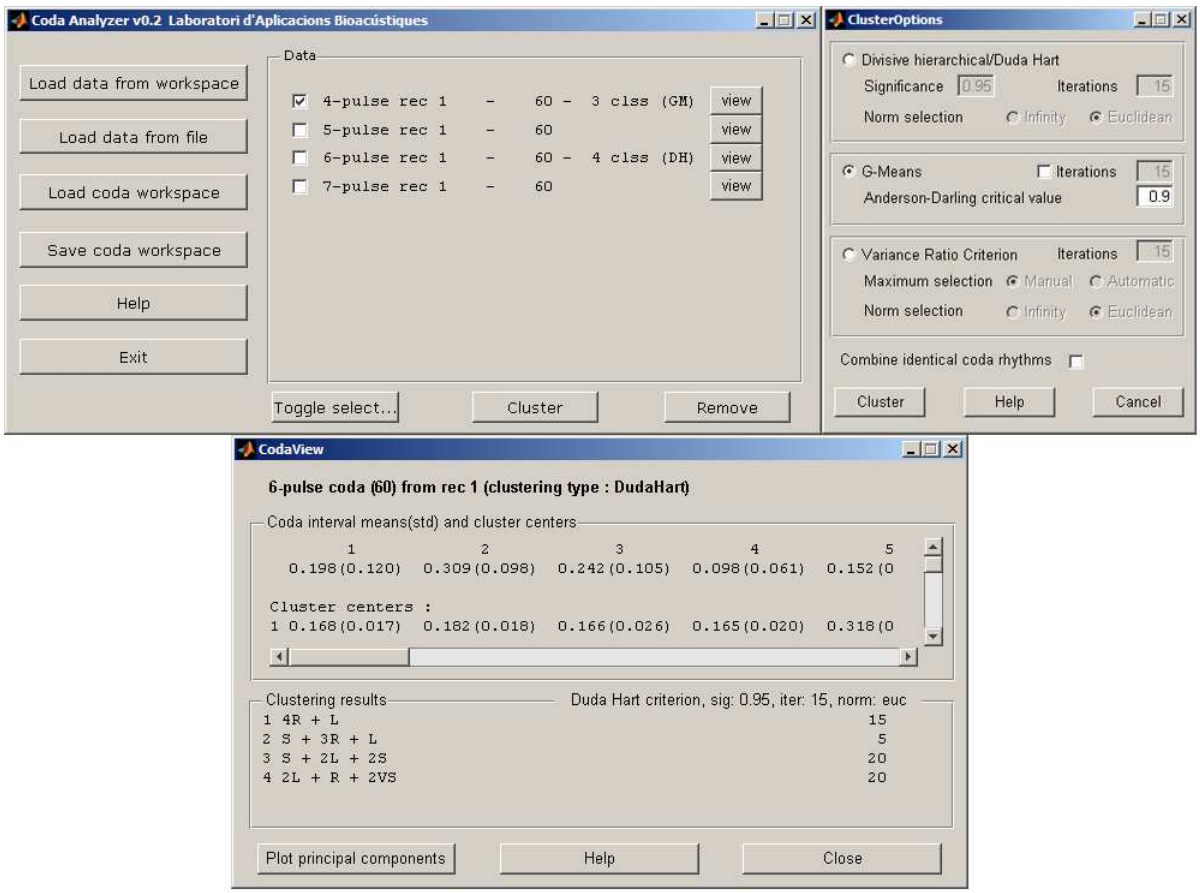

Figure 5.8: Screen shots of the coda clustering tool. Top-left shows the main screen where data can be loaded or results saved. Top-right shows the clustering options window, which allows a choice between three methods discussed in section 5.4. Bottom image shows the clustering results screen using the described labelling mechanism for the naming and including all relevant information to assess the quality of the clusters. Selecting the 'Plot' button will give a visual overview of the clusters plotted along the two principal components of the data.

[82] the coda labelling does not take into account that a single label may cover multiple classes; [73] takes into account the variability for one coda type, but does not relabel the class). As was briefly mentioned in Section 5.3, using the average of an interval to label the interval is only valid when the variance of the cluster is not too large. Otherwise, many codas in that group will actually have intervals that are much shorter or longer than the given label. This problem becomes especially apparent with the 3-click codas. Objectively, in Figure 5.6 there would only be two groups defined which both cover multiple rhythmic patterns. If more codas had been available then the gap between the two groups might had completely disappeared.

If the guidelines set out in this chapter are followed (e.g. in combination with the supplied tool) and future publications will publish inter click interval variances with their results, allowing to understand how rhythmic patterns were classified, then it will be possible to compare results between studies and to draw more definitive conclusions about the importance and uniqueness of coda rhythms for geographically separated sperm whale social groups.

There might be other techniques to classify codas based on e.g. speech recognition. A coda could be characterised as a sequence of events, which would allow it to be modelled with for example a Markov model. These algorithms can need a large amount of training data to build statistic models, which was not available for this study. Especially as there was no certainty about which coda belonged to which animal, it would be difficult to correctly train a classifier. However, once good quality data becomes available it could be interesting to analyse codas with speech recognition techniques that can take into account the sequence of pulses.

Chapter 6

Modelling of Sperm Whale Sonar through the Gabor Function

6.1 Introduction

The Gabor function [37] is described by the equation given in (6.1). It is controlled with five parameters, modelling a single central frequency f_0 and one pulse. The duration of the pulse is set through the power of the exponential damping factor α, its maximum through amplitude H and its placement and phase with t_0 and ϕ respectively. The most important reason that the Gabor function was first considered was because it had successfully been used by Kamminga and Cohen Stuart [48, 49] to model dominant frequencies of dolphin sonar. An interesting property of the Gabor function (in fact, it had been designed to satisfy exactly this property) is that it minimises the uncertainty product, the product of the temporal and spectral bandwidths. This product defines the maximum resolution that can be obtained in both time and spectral bands at the same time by specifying a lower limit on their product. This limit is only obtained by a Gabor function, any other representation will give a poorer time-frequency resolution.

$$G(t) = H \exp^{-\alpha^2 (t-t_0)^2} \cos\left(2\pi f_0 (t - t_0) + \phi\right) \tag{6.1}$$

H = amplitude, α = sharpness, f_0 = frequency, ϕ = phase, t_0 = mid epoch

An example of the Gabor model fitted on dolphin (*Tursiops Truncatus*) sonar, is presented in Figure 6.1 (the signal was recorded from *Birgitta* by Cees Kamminga in Nürnberg, 21-06-1996). As the model can only represent one frequency, the sonar signal was filtered between 50000 and 60000 Hz. The filtered signal is shown in black, with the Gabor model in green. While at the edges the fit might not be perfect, at the centre of the pulse the Gabor model represents the pulse quite good. The idea that dolphins have developed a sonar signal that is (close to) optimal in a time-frequency resolution sense is very appealing, and it gives a strong argument to investigate if other cetaceans have done the same. Of course, the application of this model is very limited here, focussing only on a single frequency, while the Gabor function can also be used for a more detailed time-frequency decomposition of the signal. From a practical viewpoint, it would be very attractive if a simple model is sufficient for identification which is why it was tested here. The next chapter will look at time frequency decompositions that will provide much more information about the signal.

In the following, the process of fitting the Gabor function on a sperm whale click will be detailed, followed by the steps necessary to prepare the data for the modelling. Then the classification method and results are presented, and finally these results and the differences with the dolphin case are discussed.

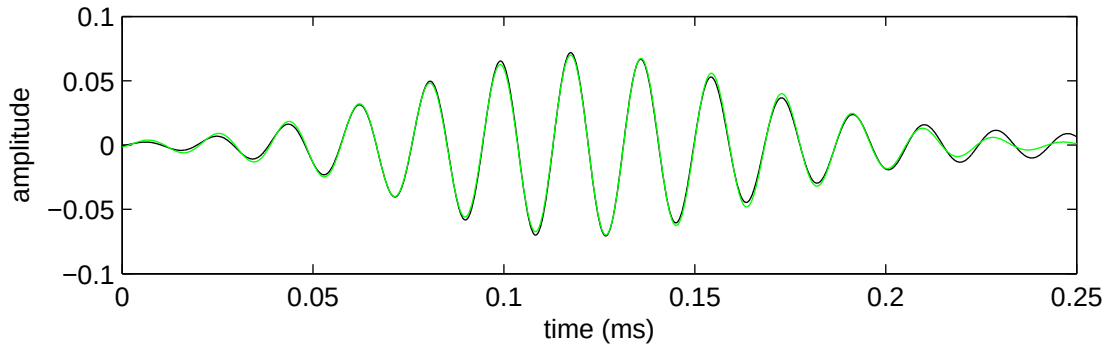

Figure 6.1: Band filtered dolphin sonar modelled with a Gabor function. Original signal is in black with the model overlaid in green. Around the peak of the signal the fit is close to perfect.

6.2 Fitting the Gabor Function

Assuming that a single frequency pulse has been extracted from a click, the Gabor function (6.1) can then be fitted on this pulse minimising the squared error function,

$$\sum_i (G(t_i; H, \alpha, t_0, f_0, \phi) - y_i)^2$$

where y_i are the points of the extracted pulse. For our data, the minimisation was performed by a standard algorithm provided by Matlab, based on the interior-reflective Newton method [23]. An initial estimate of the parameters was made in order to assure fast convergence. The amplitude H and frequency f_0 are easily estimated by the amplitude and number of zero-crossings of the pulse. The other parameters can be estimated using linear regression [14]. Writing the analytical representation of the Gabor signal and taking the natural logarithm gives,

$$\begin{aligned} G^*(t) &= \exp^{-\alpha^2(t-t_0)^2} \exp^{j(f_0(t-t_0)+\phi)} \\ \ln(G^*(t)) &= -\alpha^2(t-t_0)^2 + j(f_0(t-t_0)+\phi)). \end{aligned}$$

Then, after taking the real and imaginary parts,

$$\begin{aligned} \Re(\ln(G^*(t))) &= -\alpha^2(t-t_0)^2 = -\alpha^2\, t^2 + 2\alpha^2 t_0\, t - \alpha^2 t_0^2 = \\ &= \beta_2 t^2 + \beta_1 t + \beta_0 \\ \Im(\ln(G^*(t))) &= f_0\, t + \phi \qquad (mod(2k\pi)) \\ &= \gamma_1 t + \gamma_0. \end{aligned}$$

Using the points from the first pulse and linear regression, the above equations allow estimates of the other parameters, $\alpha = \sqrt{(-\beta_2)}$, $t_0 = \beta_1/(2\alpha^2)$ and $\phi = \gamma_0$.

6.3 Data Preparation

As explained in the introduction, the Gabor function can only be used to model a single frequency and pulse. This is of course a very limited choice. Use of the Gabor function allows a much more thorough analysis

of a signal through Gabor decomposition, but this was not yet considered here as the idea was to keep the method as simple as possible, replicating the results obtained by Kamminga on dolphins. Other methods that decompose a signal in time-frequency bands are considered in the following chapter. Considering that a sperm whale click is a broadband pulse, it is necessary to filter the signal to isolate a single frequency. Of course, this leads to the problem of selecting the frequencies that should be modelled. Research from John Goold [39] has shown that there exist a few dominant frequencies in sperm whale clicks. For male sperm whales these were found, among others, around 500 and 1500 Hz. Higher frequencies can be found, but two things have to be kept in mind. First, the directivity of a sperm whale click (see [65]) will lead to more attenuation of the higher frequency contents of a click, depending on the recording angle. Second, higher frequencies are more affected by the distance of an animal to the hydrophone. This means that a method that relies on higher frequencies may only be useful when animals are facing the hydrophone, or are in its proximity. To reduce this dependency, the focus is on the two lower frequencies presented here. Unfortunately, as lower frequencies have a longer period, they will be more affected by reverberations inside an animals head and contain pollution from other signals or echoes.

The extraction of the dominant frequencies will be demonstrated with the example click shown in Figure 6.2. As the first pulse in the signal that is modelled with the Gabor function has a short duration, only the first 10 ms of each click were considered. To obtain the dominant pulses around 500 and 1500 Hz, the clicks were band-pass filtered between 100 - 1000 Hz and 1000 - 5000 Hz. Typically, finite impulse response filters were used together with the Matlab *filtfilt* command to avoid phase distortion. After filtering, the clicks were normalised in energy. This normalisation step already makes the H parameter in 6.1 useless for classification, but it was not considered to be discriminatory from the start (as explained below, the amplitude is unreliable as a characteristic once it arrives at the hydrophone, and it was not expected that animals could be classified on their loudness).

The low frequency pulse is shown in Figure 6.3 together with its envelope. The envelope is used to isolate the pulse by taking the area of the click between two local minima of the envelope around a global maximum. In this case the area between the two dashed vertical lines includes roughly 3.5 ms and two cycles of the pulse. This should give enough points to reliably fit the Gabor model on the pulse. However, when the click is followed by a reverberation, or an echo, within a very short time interval (less than 4 ms), this method may not give enough points for an accurate estimate of the Gabor parameters.

This last problem is illustrated in Figure 6.4, where a click is almost immediately followed by an echo or reverberation. Comparing with Figure 6.2, it can be seen that the click contains many more high frequency pulses around 4 ms, which makes it less likely that a sufficient number of uncorrupted points of the first pulse can be found. Its low frequency component and envelope are shown in Figure 6.5. In comparison with Figure 6.3, the echo made the second half of the pulse unusable, not even allowing a single period of the main pulse for modelling. From the figure itself, it can be seen that the frequency estimate will be too high, as the first pulse is shortened and corrupted on the right side by the reverberation. Even when the model might fit perfectly on the reverberating pulse, it will not reflect the real frequency of the emitted signal. This effect of the changing time-delay of arrival between a click and its echoes was also shown in Figure 4.4, where the clicks could not be synchronised correctly (although synchronisation of the clicks will not be important for the application of the Gabor model).

The results of fitting the model on the click of Figure 6.2 are shown in Figures 6.6 and 6.7, for both the low and high frequency bands. Despite a slight error at the signal boundaries, the centre of the signal seems to be closely described by its model.

A summary table of the results of fitting the Gabor model on five of the data sets used for classification is shown in Table 6.1. The first two rows give the (mean) percentage of energy not explained by the model, i.e.

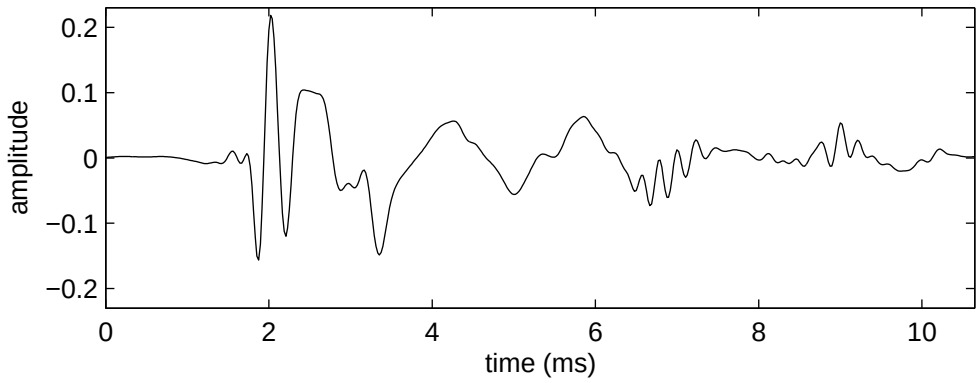

Figure 6.2: Typical example of a sperm whale click low-pass filtered at 5000 Hz.

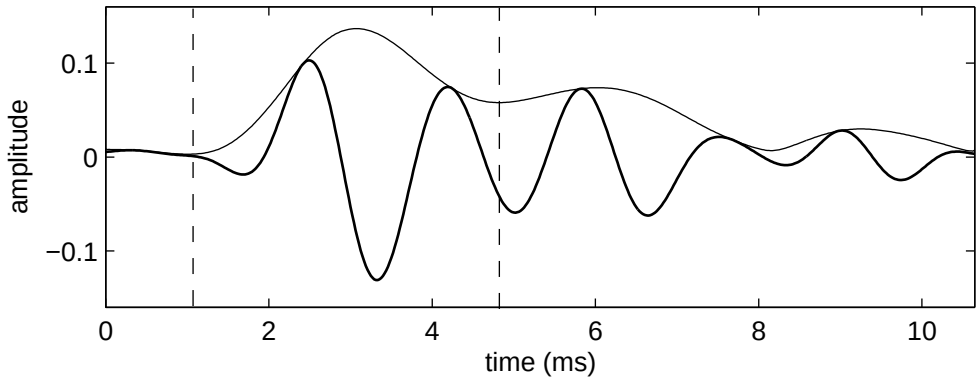

Figure 6.3: Low dominant frequency from click shown in Figure 6.2 with its envelope.

$\frac{E(signal-model)}{E(signal)}$. The last two rows give the (mean) correlation between the model and signal as a percentage. The high frequencies seem to be modelled more accurately. This can partly be explained by the reduced influence of noise (especially reverberations), and by the fact that the number of points that have to be described by the model is smaller than at lower frequencies. This can motivate the use of a higher weight when the parameters are used for identification. Only the sharpness and frequency parameters of both the low and high frequency bands were used for identification, as the amplitude and phase were considered too unreliable. The amplitude is a function of the distance and orientation of the animal to the hydrophone and cannot be considered as a characteristic feature of the animal itself. It could have value in specific situations with two animals where one is close and one far away, but it is preferred to find a parameter that can be attached to an animal rather than its distance or orientation. The phase (and t_0) will often depend on the result of the cross-correlation and is influenced by the processing of the data. On the other hand, α and f_0 can be considered to be truly characteristic of the signal itself (dependent on the selected band-pass filters) and were therefore selected as features. The class vector was constructed by taking the mean of these four features (both the low and high frequency parameters) in the training set, and classification was based on the distance of a click's characteristics to these vectors. As a purely distance based metric for classification is sensitive to the differences in magnitude and spread of the features used, the features in the training set were normalised (reduction by the mean and division by the standard deviation). The same parameters used in the training stage were then also applied to normalisation of the validation set.

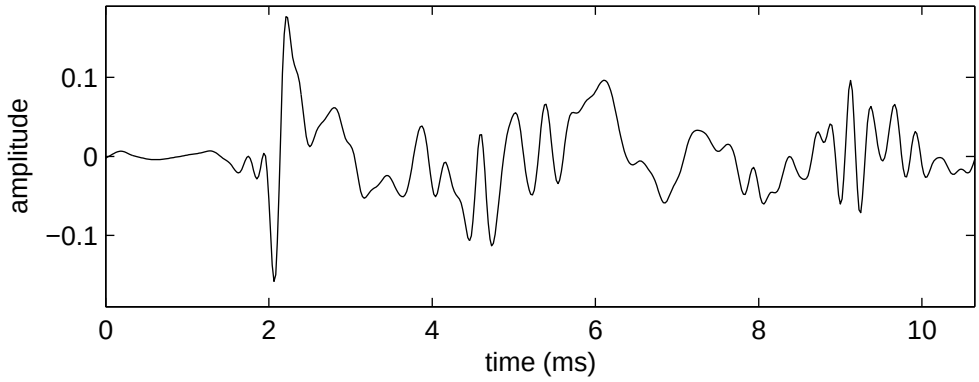

Figure 6.4: A sperm whale click low-pass filtered at 5000 Hz, taken from the same click train as the example shown in Figure 6.2, at a later point in the sequence.

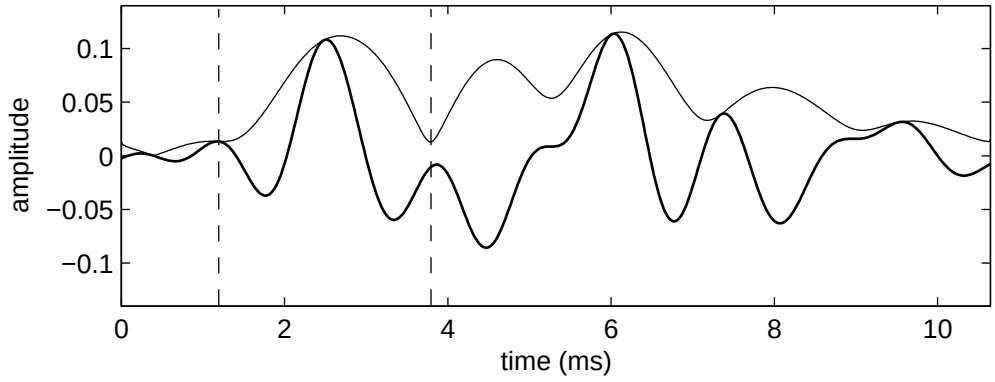

Figure 6.5: Low dominant frequency from the click in Figure 6.4. Due to a change in orientation from the animal relative to the hydrophone, a reverberation inside the head has a smaller time delay than in Figure 6.3, reducing the information available to the Gabor model.

	Click Train				
	1	2	3	4	5
energy residue % low	2.3	4.8	8.9	6.6	7.2
energy residue % high	2.1	2.3	2.1	0.7	1.2
correlation low %	98	96	94	94	95
correlation high %	98	98	98	99	99

Table 6.1: Average Gabor fit statistics. The energy residues are the averages of the percentages of energy not explained by the model. Correlation low and high give the averages of the correlation between the clicks and their Gabor models.

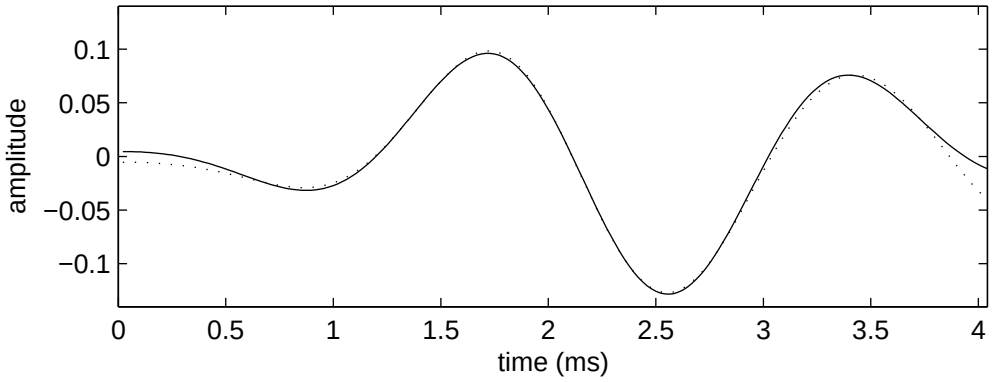

Figure 6.6: Fit of the Gabor function on the low dominant frequency (below 1000 Hz) in the click shown in Figure 6.2. The smooth line represents the original signal, with the Gabor model fit dotted.

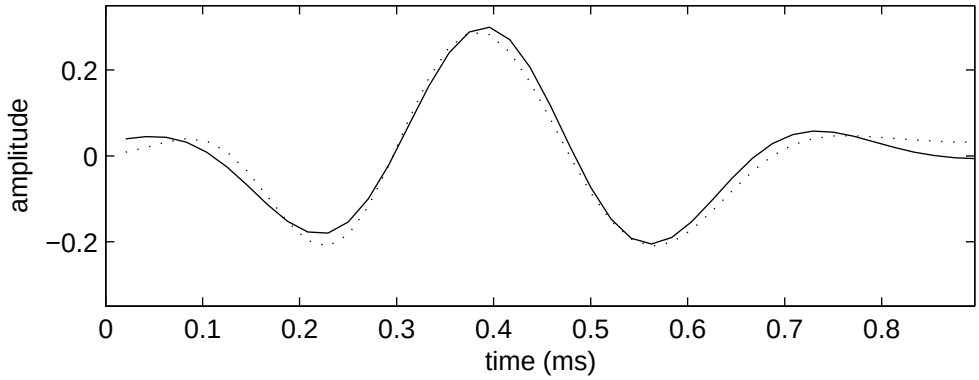

Figure 6.7: Fit of the Gabor function on the high dominant frequency (between 1000 - 5000 Hz) of the click shown in Figure 6.2. Lines are as in Figure 6.6.

6.4 Classification with the Gabor Function

To assess the performance of the selected Gabor function parameters as characteristic features, five data sets were selected (without using any selection criterion, except that the complete dive was not considered for a first test) for classification. To obtain the class vectors, the first 50 clicks of each set were used for training, with the remainder used for validation. The class vectors were defined as the mean of the features in the training set. The values of the training features with their centres are shown in the two scatter plots in Figure 6.8. It can be seen that there is some clustering that could allow discrimination between the animals, but the clusters are packed together densely with overlapping samples. The results of classification using all four features are summarised in Table 6.2. The training data already shows poor performance, especially the fifth set would not be reliably classifiable. The validation set might show a preference for the third class, and overall it can be concluded that the four parameters cannot directly be used for classification.

Looking at the left graphic of Figure 6.8, to see how classification of the fifth animal (blue) can be improved, it does show both a large spread in the dimension of the low frequency α, combined with a large overlap with other classes. Removal of the low frequency α gives the classification results shown in Table 6.3, which shows the classification of the fifth animal greatly improved for both the training and validation sets, at the cost of generalisation of especially the third animal. The high frequency α also shows a large spread, but removal of this feature worsens the classification as a whole.

From here, several improvements are possible. A first step could be finding weights for the four Gabor parameters, as a zero-weight for one of them showed an improvement which is shown in Table 6.3. Especially the performance on Set 5 was significantly improved, without reducing results in the other classes. But the spread of the data in the feature space does not look very promising by itself, and some transformation of the feature space might be required to create distance between the classes. Unfortunately, the Gabor model does not easily allow to increase the number of usable features; a higher dimension might lead to better separation as well. These kind of improvements were not further researched at this point, an improved feature selection algorithm will likely lead to better results more quickly.

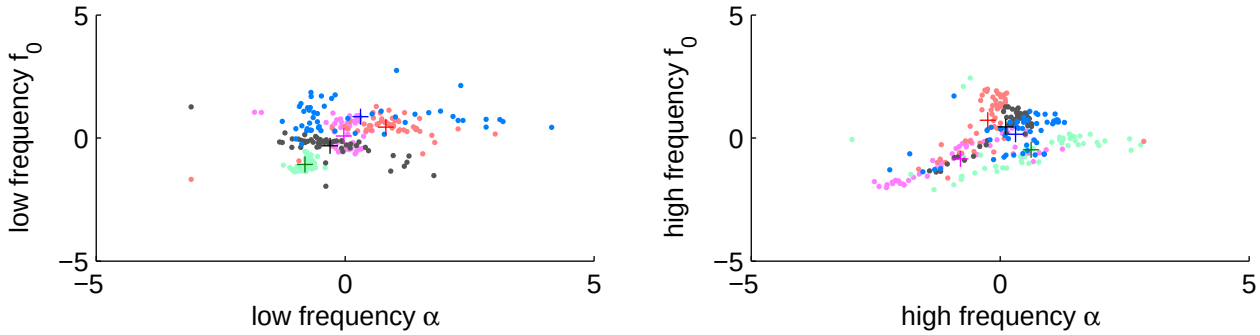

Figure 6.8: Scatter plot of the two pairs of features selected for classification from the Gabor parameterisation. The large marks represent the centres of the individual features, used for the construction of the class vectors.

6.5 Conclusion

While the modelling of the dolphin sonar signal using the Gabor function was successful in other studies, this model had never been used for marine mammal classification. Dolphin signals are usually recorded

	Training set					Validation set				
Classified as	1	2	3	4	5	1	2	3	4	5
Set 1	58	18	20	18	12	44	24	2	4	2
Set 2	10	78	2	2	0	6	39	3	19	7
Set 3	0	0	76	8	28	7	0	71	26	48
Set 4	2	4	0	72	26	22	24	18	51	4
Set 5	30	0	2	0	34	21	12	6	0	39

Table 6.2: Click classification based on modelling two dominant frequencies with the Gabor function, values are correctly classified percentages.

	Training set					Validation set				
Classified as	1	2	3	4	5	1	2	3	4	5
Set 1	58	20	26	18	10	42	15	3	4	4
Set 2	12	76	2	2	0	12	61	3	21	7
Set 3	0	4	68	2	10	1	0	50	3	2
Set 4	0	0	2	76	8	29	21	36	71	7
Set 5	30	0	2	2	72	16	3	8	0	80

Table 6.3: Click classification based on modelling two dominant frequencies with the Gabor function, values are correctly classified percentages.

on-axis and from close distances, leading to very high quality recordings (generally, low quality recordings are simply discarded). Sperm whale recordings on the other hand are practically never on-axis and usually of low quality, with the animal at a large distance and high background noise levels. If recordings were available with clicks as shown in Figure 4.2, then fitting the Gabor function might have been very successful as well. As it is, the multiple pulses in the off-axis clicks are making the Gabor function unsuitable to be used as a reliable model. When the change in time-delay between a pulse and its reverberation leads from a situation as shown in Figure 6.3 to the one shown in Figure 6.5, a sudden jump in the values of the features can be expected. It will be difficult for a classification algorithm to take this into account.

It is possible that weighing the features, using some transform of the feature space, or a different classifier altogether, will improve the classification. However, the images in Figure 6.8 do not immediately suggest an improved classification approach and the features themselves are very dependent on the quality of the Gabor function parameterisation, which proved unreliable. Instead of trying to improve the classification, at this point it could be more fruitful to improve the feature selection procedure itself, moving away from the strict framework of the Gabor model. The next chapter will present a feature extraction method that is more flexible and especially fine-tuned to the data at hand.

Chapter 7

Feature Selection

7.1 Introduction

As was shown in Chapter 6, modelling a single frequency does not characterise the clicks of an individual animal very well. Assuming that there do exist individual acoustic cues in a click, these will be somewhere in the time-frequency domain. There are many algorithms (short-time Fourier transform, instantaneous frequency, cosine transform, wavelets, etc) that allow the decomposition of a signal in time-frequency components with different bandwidth trade-offs. As it is not clear in advance what kind of feature is sought, it is impossible to favour one algorithm over another without extensive testing. What is important is that the method allows flexibility in the choice of the representation, to easily adapt to a different situation without having to reevaluate different algorithms.

Wavelet packets give this kind of flexibility (Section 7.2), as they give a redundant representation of a signal, with ample choice between time-frequency bandwidth trade-offs. When all signals are expressed in a wavelet packet basis, within-class and between-class differences can be compared between all available coefficients to find a basis that allows maximum separation between classes (Section 7.3). The usual requirement that the basis should be able to reconstruct the original signal could be relaxed for this application, as there is no interest in reconstruction after the features are extracted. Once a basis is selected, a number of individual coefficients can be chosen from that basis to separate the classes (Section 7.4). For the course of this thesis, wavelet transforms other than discrete filter banks were not considered. While e.g. complex continuous wavelet transforms might also be good candidates to find differentiating features (e.g. [31]), these are also computationally much more intensive than the simple filter-bank structure of the discrete wavelet transform. As real-time performance is considered one of the most important requirements of the final algorithm, the focus was aimed at the discrete transform.

A different time-frequency decomposition may be able to characterise differences between classes in less coefficients than when using wavelet packets, it is expected that the same information is available in the packet decomposition and the subsequently selected coefficients. The use of the redundant representation and flexible selection of features will then ensure that the algorithm is not too biased towards a specific data set.

7.2 Creation of Wavelet Packet Decomposition

From a signal processing point of view, a wavelet transform will split a signal in two frequency bands using a high- and low-pass filter, keeping the high frequency wavelet coefficients and re-splitting the low frequency

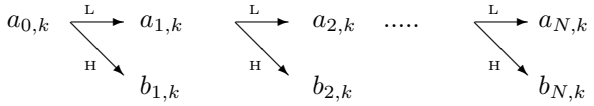

Figure 7.1: Discrete wavelet transform through a low-pass and high-pass filter. The signal $a_{0,k}$, where k is the coefficient index, enters at the left. The filter creates the downsampled scale ($a_{1,k}$) and wavelet ($b_{1,k}$) coefficients. The scale coefficients are then run through the filter again until the lowest scale has been reached. The decomposition in dyadic frequency bands is shown in Figure 7.2.

scaling coefficients. A schematic of the discrete wavelet transform (DWT) is shown in Figure 7.1. The signal $a_{0,k}$ enters the filters, which produce the new coefficients at a lower scale $a_{1,k}$ (low-pass) and wavelet coefficients $b_{1,k}$ (high-pass). The subscript k is the index of the coefficient, which in our case ran from 0 to 511. The outputs of these filters contain some redundancy as there are now 2*512 samples. To remove this redundancy, the filter outputs are down sampled by two, resulting in 2*256 samples.

The corresponding equations are given as

$$a_{j+1,k} = \sum_l c(l-2k)a_{j,l} \quad \text{and} \quad b_{j+1,k} = \sum_l d(l-2k)a_{j,l},$$

where $c(n)$ are the low-pass and $d(n)$ the high-pass filter coefficients; $a_{j,l}$ is coefficient l of the signal at filter step j. Note the recursive nature of these filters, as the output $a_{j+1,k}$ is the input of both filters in the next filter step. The down-sampling by two operation is expressed in the $2k$ term. The connection between these filters and wavelets is given by

$$\phi(t) = \sum \sqrt{2}c(n)\phi(2t-n) \quad \text{and} \quad w(t) = \sum \sqrt{2}d(n)\phi(2t-n).$$

The equation on the left is called the dilation equation, and the one on the right the wavelet equation. It can be seen that, in the case of a quadrature mirror filter bank, the low-pass filter coefficients $c(n)$ directly define both the dilation equation and its corresponding wavelet. The discrete wavelet transform is calculated recursively by the filter equations, the scaling coefficients $a_{j+1,k}$ by the low-pass filter, and the wavelet coefficients $b_{j+1,k}$ by the high-pass filter until a lowest scale is reached. The signal is then represented by the wavelet coefficients $\{b_{i,k}\}_{i=1}^{N}$ and the lowest scale coefficients $a_{N,k}$.

The difference between the normal wavelet transform and the wavelet packet algorithm is that the latter will continue entering both outputs $a_{j,k}$ and $b_{j,k}$ back into the filters, storing all (down sampled) coefficients at every level. Figure 7.2 shows how frequency bands are dyadically decomposed every time the coefficients are run through the filter. A standard discrete wavelet transform would only contain the 0-3 ($a_{4,k}$), 3-6 ($b_{4,k}$), 6-12 ($b_{3,k}$) and 12-24 kHz ($b_{2,k}$) intervals (bins). The entire wavelet packet table gives a redundant representation of the signal, and many different bases can be selected from it to rebuild the original. For example, from Figure 7.2, a possibility would be the 0-6, 6-9, 9-12 and 12-24 kHz intervals, another choice could be 0-3, 3-6, 6-12, 12-18 and 18-24 kHz. For identification the interest is in finding a basis that will emphasise the differences between classes, which is discussed in the next section. Although creating complete wavelet packet tables might be time consuming, once the basis is chosen only a small part of the tree has to be rebuilt for the classification process.

The filter coefficients $c(n)$ used for the classification of sperm whales correspond to the Symmlet wavelet, which has the property that it is almost symmetrical. This almost symmetry gives a minimal phase distortion to the filtered signal. The frequency response of a Symmlet wavelet filter bank is shown in Figure 7.3. It was created using a Symmlet with 8 vanishing moments, which was initially used for the wavelet packet construction. The image shows the low-pass filter, but the response of the high-pass form is simply mirrored. It can be seen that the phase delay is almost constant. The roll-off of the filter is slow, which combined with down-sampling will give aliasing. Normally, this is cancelled again in the synthesis filter bank to give perfect reconstruct. In fact, the coefficients of the quadrature mirror filters are computed based on the requirement to cancel both aliasing and phase distortion. However, for the purpose of discrimination between animals the signals will not be reconstructed and aliasing is not a problem as long as it is consistent in all patterns; there is no interest in true energy levels in specific frequency bands in the signal.

split level	Frequency Bands							
1	0-24 kHz							
2	0-12 kHz				12-24 kHz			
3	0-6 kHz		6-12 kHz		12-18 kHz		18-24 kHz	
4	0-3 kHz	3-6 kHz	6-9 kHz	9-12 kHz	12-15 kHz	15-18 kHz	18-21 kHz	21-24 kHz

Figure 7.2: Wavelet packet table, showing the decomposition of the frequency bands every time the signal is sent through the filter. For example, bin $(4, 2, \cdot)$, containing the 3-6 kHz band, corresponds to the wavelet coefficients $b_{4,k}$ in Figure 7.1. However, bin $(4, 3, \cdot)$, covering the 6-9 kHz band, corresponds to the scale coefficients obtained after re-filtering the wavelet coefficients $b_{3,k}$ and are not normally available in the discrete wavelet transform.

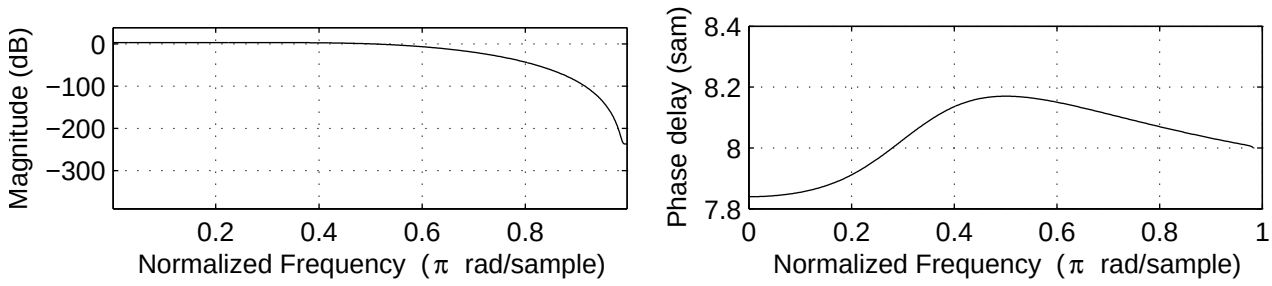

Figure 7.3: Frequency response and phase delay in samples of a Symmlet quadrature mirror filter bank with 8 vanishing moments.

7.3 Selection of a Local Discriminant Basis

The discriminating basis construction is described in detail in [84]. After a WPT has been generated for every click in the training set, a representation of the signal is searched that maximises the distance between the different classes. One way to measure the difference between classes is to measure the difference in energy in a specific time-frequency bin. In order to do this, the WPTs of the clicks from one class have to be combined. This is achieved with a time-frequency energy map, defined as

$$\Gamma_c(j,k,m) = \sum_i^{N_c} (\hat{x}_i^c(j,k,m))^2 / \sum_i^{N_c} ||\mathbf{x}_i^c||^2$$

where (j,k,m) denotes the position in the WPT, at splitting level j, frequency band k and coefficient m within the bin (for example, in Figure 7.2, the position $(4,5,6)$ corresponds to the sixth coefficient in the 12-15 kHz frequency band); $\hat{x}_i^c(j,k,m)$ denotes the wavelet coefficient of click sample i and class c at position (j,k,m); $\mathbf{x}_i^c$ the original click sample i of class c; N_c the number of training samples in class c.

After all the individual tables for each click have been collapsed into one energy map per class, the discriminating power of a specific bin $(j,k,\cdot)$ can be measured by summing the differences of the coefficients in the bin between every pair of classes,

$$\mathcal{D}(\{\Gamma_c(j,k,\cdot)\}_{c=1}^C) = \sum_{p=1}^{C-1} \sum_{q=p+1}^{C} \mathcal{D}(\Gamma_p(j,k,\cdot), \Gamma_q(j,k,\cdot)).$$

In the above expression $\mathcal{D}$ is an additive discriminant measure, for which we used the squared l^2-norm; C is the total number of classes. A high value for $\mathcal{D}$ indicates that coefficients between at least two classes lie far apart, and that the bin may be able to differentiate between at least those two classes.

The discriminating basis can now be constructed with the following rule, if the discriminant measure over a bin, $(j,k,\cdot)$, is higher than the sum of the measures taken over the two bins it splits into, $(j+1,2k,\cdot)$ and $(j+1,2k+1,\cdot)$, then the 'parent' bin $(j,k,\cdot)$ is selected, otherwise it is split up. For example, using Figure 7.2, if $\mathcal{D}$('0-12 kHz') $>$ $\mathcal{D}$('0-6 kHz') + $\mathcal{D}$('6-12 kHz'), then the frequency band 0-12 kHz is selected, otherwise the two half-bands are used. It should be noted that it is important that time-frequency bins that vertically overlap in Figure 7.2 are never selected at the same time, not only because the information would be redundant for reconstruction of the signal, but also because the coefficients will be strongly correlated as the wavelet filter is a linear operator and thus the information would also be redundant for discrimination.

7.4 Classification based on LDB coefficients

Extraction of the most discriminating features of the sperm whale clicks was first done on the same classes that were used for classification using Gabor function parameters in Chapter 6. To ensure that low frequency artefacts and disruptive sea-noise were ignored, the data was band-pass filtered between 100 and 20000 Hz using a fifth order Butterworth filter. Furthermore, a wavelet based denoising algorithm was used to reduce noise, applying a soft threshold on the wavelet coefficients [30]. This algorithm reduced the wavelet coefficients by a factor of the standard deviation measured on the noise levels in each recording. The fifteen strongest coefficients were selected for classification according to Fisher's discriminant given by :

$$FD = \frac{\sum_c \left(\overline{s}_i^c - \text{mean}_c(\overline{s}_i^c)\right)^2}{\sum_c \text{var}_i(s_i^c)} \tag{7.1}$$

where s are coefficients taken from a specific entry in the discriminating basis, and both the bar and var_i take the mean and variance over all samples s_i in class c and mean_c takes the mean over all classes. Essentially, this expression measures the distance between the class means and their common centre with respect to their widths, leading to high values when samples in a class lie tightly around their class centre.

The number of features was chosen fairly arbitrarily at this point, with the initial objective to see if the wavelet coefficients were suitable at all for discrimination. The influence of the number of features will be

looked at in the next chapter. The same approach was used as with Gabor, the class vectors were created by averaging the features of the first 50 clicks (after all clicks were normalised by their energy). The remainder of each set was then used for validation. The length of the clicks was 512 samples, around 10 ms. The wavelet decomposition was done with a Symmlet wavelet with 8 vanishing moments (16-sample support). The wavelet packet table was created using 4 frequency splits, giving at the lowest decomposition level time-frequency bins containing 32 samples and covering a frequency band of 1500 Hz. Splitting the signal further would make the output unreliable as the signal and filter lengths would have been equal. To have an idea of the variability of the features a plot was made of the evolution of the two strongest features during the dive in Figure 7.4. It can be seen that there are both intervals with smooth trends as some seemingly random behaviour, perhaps due to background noise or directivity issues. Based on this figure, using 50 consecutive clicks might be just enough to describe the variability of the feature.

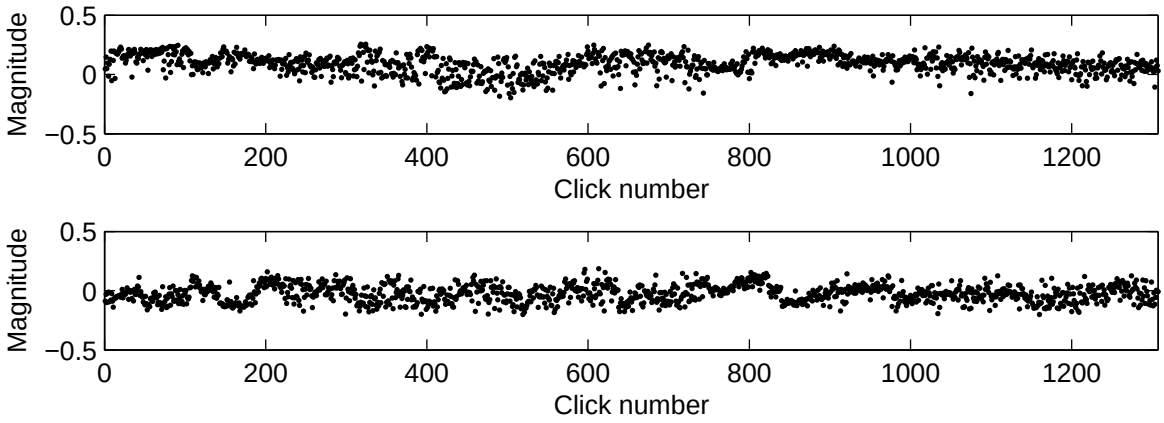

Figure 7.4: Variability of the two strongest discriminating features (first feature on top, second on bottom) during the complete dive.

The selected features are summarised in Table 7.1. What becomes immediately apparent is that the coefficients are focussed on the lower frequencies in the signal, where multiple dominant frequencies can be found according to Goold [39]. Using the same features for classification gives the result shown in Table 7.2. The training data on the left side of the table can be classified quite successfully, much better than compared to the results from Tables 6.2 or 6.3. However, the performance on the validation set shows that the generalisation for the last three sets is very poor.

Noting from Table 7.1 that the algorithm exclusively uses low frequency components, an improvement in the feature selection might be found by low-pass filtering the data. While wavelets already split up the frequency bands, the cut-offs are certainly not sharp (as was shown in Figure 7.3) and the high frequencies may add noise to the selected low frequency coefficients. Otherwise, if the classification itself does not perform worse, being able to focus on a smaller bandwidth and to down-sample the data will speed up the algorithm. For the low-pass filter a fifth order Butterworth filter was used (an infinite impulse response filter was preferred due to the short data segments that are analysed, a finite impulse response filter generally needs more coefficients to reach comparable performance); it is good to look at its characteristics before continuing. The frequency response is shown in Figure 7.5. Butterworth filters [20] have the property that they are maximally flat, but with a slow roll-off, as can be seen in the magnitude response graphic on the left. When down-sampling this should be taken into account, and the Nyquist frequency should not be too close to the cut-off frequency to avoid strong aliasing.

index	split	frq band (Hz)	power	index	split	frq band (Hz)	power
1	5	1 - 1500	3.4	2	5	1 - 1500	3.5
3	5	1 - 1500	3.0	4	5	1 - 1500	2.3
9	5	1 - 1500	1.0	10	5	1 - 1500	0.80
13	5	1 - 1500	0.79	14	5	1 - 1500	0.76
16	5	1 - 1500	0.65	18	5	1 - 1500	0.79
21	5	1 - 1500	0.95	22	5	1 - 1500	0.73
31	5	1 - 1500	1.2	40	5	1500 - 3000	0.80
43	5	1500 - 3000	1.3			-	

Table 7.1: Selected wavelet packet coefficients for discrimination. The given index is the position of the coefficient in the memory array holding all coefficients of the signal at the same splitting level. The splitting level starts at 1, which indicates the original signal, i.e. at level 5 the signal was filtered 4 times. The frequency band should be considered approximate as the filters are not 'brick walls'.

	Training set					Validation set				
Classified as	1	2	3	4	5	1	2	3	4	5
Set 1	100	0	0	0	0	100	3	1	0	0
Set 2	0	100	2	8	0	0	97	5	3	44
Set 3	0	0	82	18	0	0	0	32	13	0
Set 4	0	0	10	62	0	0	0	22	61	0
Set 5	0	0	6	12	100	0	0	40	23	56

Table 7.2: Click classification based on the 15 most discriminating wavelet coefficients in Table 7.1.

Using Table 7.1 as a guide, almost all features seem to be found below 2000 Hz, which was taken as the cut-off frequency for the Butterworth filter. The data was decimated three times leaving a bandwidth up to 3000 Hz. Going back to Figure 7.5, the filter also has a considerable phase-delay at the low frequencies. This is not necessarily a problem for classification, since the shift would be consistent for all signal patterns. However, it can lead to low frequency characteristics being pushed out of the analysis window. For this reason, the Matlab *filtfilt* command was used for filtering operations, instead of *filter*. The former command filters the signal twice, the second time with the time-reversed signal, to undo the phase shift. The difference between the commands is demonstrated in Figure 7.6. The original signal (100 - 5000 Hz) is shown in blue, with a clear low frequency component around sample 170. The green line is filtered normally, where the low frequency was delayed up to a 100 samples further, around sample 270. The red line was filtered twice and corresponds precisely to the original signal.

The patterns were filtered and down-sampled as described above, and the discriminating features were selected again, leading to the features in Table 7.1. To take into account the much shorter patterns, the package table only used 2 splits, and a Symmlet wavelet with 5 vanishing moments. There appears one feature in the table from the frequency band above 2000 Hz, which was attenuated by the low-pass filter. It shows the slow roll-off of the Butterworth filter, and might indicate some aliasing. Classification results are shown in Table 7.4, and it could be argued that the over-all performance compared to Table 7.2 has been slightly improved. While the second and fifth classes classified worse, classes three and four were improved by larger margins. But this would be somewhat suggestive, what could be concluded is that performance was not worse, while much less information (fewer samples per click) was used. This would be preferable for a real-time implementation of the algorithm.

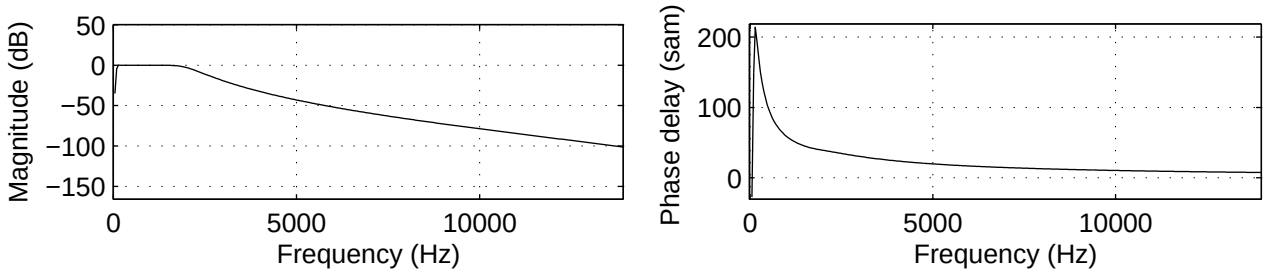

Figure 7.5: Frequency response and phase delay in samples of a fifth order 100 - 2000 Hz band-pass Butterworth filter.

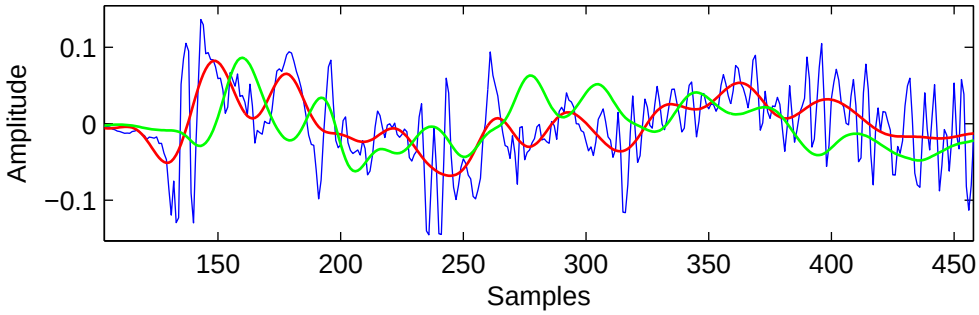

Figure 7.6: Illustration of the phase shift difference between Matlab *filter* (green) and *filtfilt* (red) commands compared to the original signal (blue).

index	split	frq band (Hz)	power	index	split	frq band (Hz)	power
1	3	1 - 750	3.4	2	3	1 - 750	3.4
3	3	1 - 750	4.1	6	3	1 - 750	1.2
8	3	1 - 750	1.0	9	3	1 - 750	1.0
10	3	1 - 750	1.3	12	3	1 - 750	0.89
16	3	1 - 750	2.4	20	3	750 - 1500	1.2
22	3	750 - 1500	1.7	23	3	750 - 1500	1.1
25	3	750 - 1500	1.4	26	3	750 - 1500	1.1
52	3	2250 - 3000	1.0			-	

Table 7.3: Selected wavelet packet coefficients for discrimination using data that was band-pass filtered between 100 and 2000 Hz. It is noted that the last coefficient falls outside the filter's pass-band.

	Training set					Validation set				
Classified as	1	2	3	4	5	1	2	3	4	5
Set 1	100	0	0	0	0	100	3	3	1	0
Set 2	0	100	4	0	2	0	88	7	1	2
Set 3	0	0	92	0	8	0	0	61	0	61
Set 4	0	0	4	100	2	0	9	29	96	4
Set 5	0	0	0	0	88	0	0	0	1	33

Table 7.4: Click classification based on the 15 most discriminating wavelet coefficients in Table 7.1.

7.5 Conclusion

In this chapter the suitability was investigated of using wavelet coefficients as discriminating features for identifying sperm whales. It was found that a significant increase in performance can be obtained compared to using a Gabor function model to describe sperm whale clicks. The creation of a discriminant basis and subsequent selection of most discriminating features showed a special interest for low frequency characteristics, in line with earlier research from Goold [39] who found multiple dominant frequencies for male sperm whales below 2000 Hz. While this can not immediately be generalised to all sperm whales, a cut-off frequency around 3000 Hz might be good enough to include females, as their dominant frequencies were found to be somewhat higher. In practice, the creation of the local discriminant basis will likely be based on the whole frequency band. But as was explained in Section 7.4, the change in phase delay of pulses within the click may lead to noise propagating through the wavelet packet table. Therefore, after it has been confirmed that most interest is in the lower frequencies, subsequent patterns can all be low-pass filtered and down-sampled. This is especially interesting for real-time, embedded applications, as it will result in faster execution of the classification process, leading to lower CPU and battery power requirements.

The classification itself has only been demonstrated on a sub set of the data. Tables 7.2 and 7.4 showed good performance on only three of the five sets used in the classifier, although especially Table 7.4 had very good results on the training data. As these features showed promising results, the selection procedure itself was not further investigated. Perhaps a different wavelet, a translation invariant wavelet transform, or, say, a different metric to measure the distance between bins in the wavelet packet table could lead to improved results (not to mention other time frequency representations). The number of parameters that can be changed or fine-tuned is large, but there is no immediate indication of which one could significantly improve results. Translation-invariant transforms were considered, as these could perhaps correct energy shifts introduced by errors or problems in the synchronisation routines. A single sample error here would mean that all energies in the inputs (and outputs) of the wavelet filter bank are shifted as well, possibly leading to the selection of a wrong wavelet coefficient. But the coefficient selection in Tables 7.1 and 7.3 showed a preference for coefficients with as little precision in time as possible (i.e. at the lowest splits in the wavelet packet table (Table 7.2) with narrow frequency bandwidth but maximal spread in time). It could be argued that this is precisely because of errors introduced during synchronisation, but keeping in mind Figures 4.4 and 4.5, the patterns will never really line-up exactly, and features that have less precision in time can always be expected. The good performance from selected coefficients with maximal spread in time indicated that a translation invariant wavelet transform might not have added much to the results, while at the same time it would have been rather expensive in terms of computational complexity. Therefore, its application was not researched in-depth. However, it should be kept in mind as a possible improvement in situations where synchronisation completely fails (errors larger than just a few samples).

Considering the options for improvement in the feature selection itself, and the improvement that could be gained by selecting a classifier that is capable of modelling the feature space more accurately than was done in this chapter, it was decided to concentrate on the latter suggestion.

Chapter 8

Non-linear Classification Approach

8.1 Introduction

The previous two chapters used a simple linear classifier to recognise the individual sperm whales. This performance probably could have been improved, for instance, by weighing the input features (e.g. using a perceptron model). But these kind of linear classifiers may have other problems, such as global behaviour where features that are at a large distance from class centres are still classified in that particular class. Another type of classifier might lead to better improvement easier, and to assess this it is important to obtain some insight into the feature space. Figure 8.1 shows two scatter plots of four features from five animals. The features were the strongest ones according to Fisher's power of discrimination (7.1) and taken from the local discriminant basis of the first 50 clicks that were used for the training sets. Two things can be noted from the plots; first, the features show a fairly strong clustering tendency (certainly much better than was found for the Gabor function parameters in Figure 6.8). The black animal in the right plot has considerable spread without a clear cluster centre, but the other classes do seem to be packed around a centre. A second point is that the combination of these four features separate the five animals reasonably well. Where in the left plot the yellow and green animals have considerable overlap, in the right plot they are separated (although close together). Likewise, in the right plot black and blue animals are overlapping (with poor clustering from black), but in the left plot they are completely separated (with improved clustering from the black animal).

To exploit the clustering tendency of the features shown in Figure 8.1, the feature space can first be modelled by placing a number of clusters on top of the locations that have highest densities. This can be done with a clustering algorithm. Once such a model is available, weights can be assigned to each cluster to indicate its importance for each class. The weighed distance of a new pattern to the clusters will then decide its class. Classification based on a model built from clusters and weighed distances is precisely what a radial basis function network (RBF) can be trained to do, and it forms a natural choice as classifier. The benefit of this type of network is that while the clustering process can be time-intensive, this only needs to be done once on the available training data. After the model parameters are known, training of the remainder of the network goes very fast (due to the low number of classes that need to be separated for sperm whales), as is the classification process itself.

The RBF approach focussed on the centres of the clusters in the feature space. Another approach is to concentrate on the boundaries between classes, which is done by support vector machines. SVM have a reputation of good generalisation and considering the limited training data that will be available to classify the sperm whales this is certainly a useful property. While the SVM architecture is identical to RBF, the

training process is very different. SVM optimises both the number of centres and the output weights together. This can make the complete classification process operate more slowly, but the identical structure made them an obvious candidate for testing and for comparing the results obtained with the RBF network.

In the following, the RBF network will first be discussed starting with clustering of the data (Section 8.3.1). Clustering was done both with k-means, which needs to know the number of clusters in advance, and a variant (adapted c-means as discussed in Section 5.2) which accepts normally distributed classes, and does not require prior knowledge of the number of clusters. Then the use of the RBF network and its training procedure are explained (Section 8.2), followed by selection of distribution parameters and classification of all the data (Section 8.3). Finally, SVM will be discussed with a brief description of the algorithm (Section 8.4.1), selection of distribution parameters and classification (Section 8.5.1).

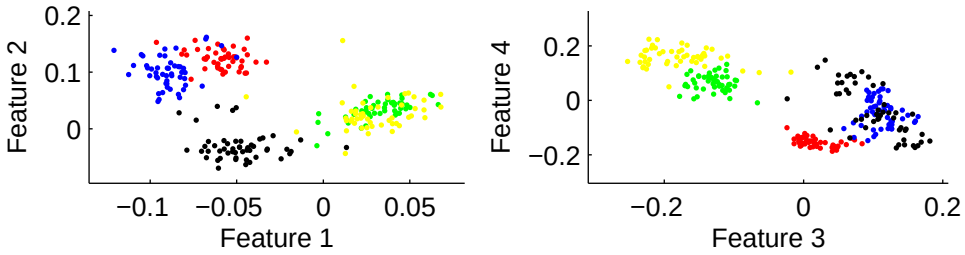

Figure 8.1: Scatter plots of four features from five animals; the features were taken from the first 50 clicks used for training the algorithms.

8.2 Radial Basis Function Networks

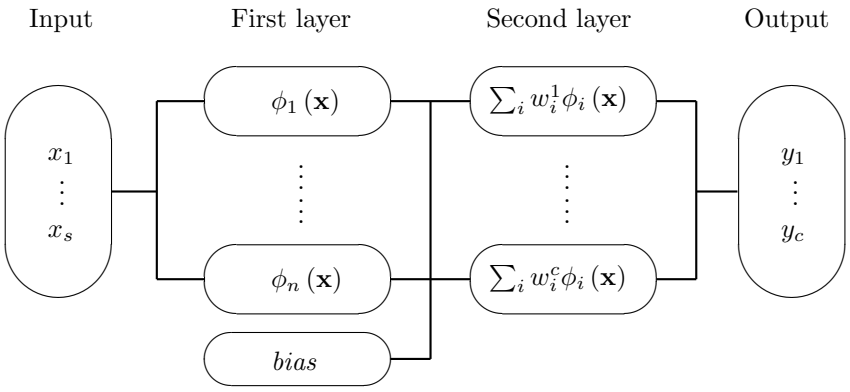

Figure 8.2: Schematic of an RBF-network as described by (8.1). An s-dimensional sample $\mathbf{x}$ enters on the left. It is first run through the n hidden layer nodes, where the distances between $\mathbf{x}$ and centres $\mathbf{c}_i$ are evaluated through Gaussian functions $\phi_n(\mathbf{x})$. The outputs of the n Gaussian functions are then weighed with weights w_i^j and linearly combined in the second layer nodes (containing one node per class). An additional weight (w_0^j) is usually added to each second layer node to account for the bias factor. This bias is then represented by an additional constant activation function in the first layer, $\phi_0 \equiv 1$. Taking the output vector $\mathbf{y}$, the class of the sample $\mathbf{x}$ is computed by $\arg\max_i \; y_i$.

Radial basis function networks [15] consist of two layers, a hidden layer and an output layer. A single output of the network in generic form with N nodes in the hidden layer can be written as :

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\|\mathbf{x} - \mathbf{c}_i\|). \tag{8.1}$$

A schematic of the network is given in Figure 8.2. Each output is a weighed linear sum of non-linear activation functions $\phi : \Re^N \rightarrow \Re$ in the hidden layer, with as input the norm between a feature vector $\mathbf{x}$ and centres $\mathbf{c}_i$. The norm makes clear that RBF networks are distance based, where the Euclidean norm is most popular. The activation function is a radial function, which has the property that it monitonically increases or decreases with distance from a central point. A common choice that was also followed in this work is a Gaussian kernel, defined by

$$k(\mathbf{x}, \mu) = \exp(-\frac{||\mathbf{x} - \mu||^2}{2\sigma^2}), \tag{8.2}$$

where $\mathbf{x}$ is the input feature vector, μ controls the kernel's centre, and σ its width.

Various methods can be used for training the network, a list of approaches can be found in [90]. Taking note of the strong clustering displayed in Figure 8.1, one method using unsupervised clustering suggested in [67] seems especially suited. Using this approach the two layers in the network each get different tasks. The hidden layer is used to model the feature space, and the output layer performs the classification. This gives the advantage that the network layers can be trained separately, without the use of a time-consuming back-propagation algorithm as e.g. with perceptron networks.

Training of the first layer can be done by placing Gaussian kernels on the data, for example through k-means clustering. This clustering can give both the mean and standard deviation parameters of the kernels, but not the optimal number of kernels needed. When the training and evaluation of a network goes fast then k-means can be tried for several different values, as was done in [96], but there are various clustering methods that can search for an optimal (according to some defined statistic) number of clusters. One example of such an algorithm is the adapted c-means that was described in Section 5.2.

Training of the second layer then becomes a fast and linear process. The number of nodes in the second layer is set to the number of classes, using binary encoding for the targets. This means that, in the case of three classes, a sample from class 1 has target $[1\ 0\ 0]^t$, and a sample from class 3 has target $[0\ 0\ 1]^t$. Once the hidden layer has been trained, the second layer weights can be calculated directly as follows [15]; writing the network output as

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}\ \mathbf{k}, \tag{8.3}$$

where $\mathbf{W}$ is the weight matrix with weights w_{ij} (the weight at output node j belonging to the kernel function i), and $\mathbf{k}$ a vector with the hidden layer kernel outputs $k_i(\mathbf{x}, \mu)$. The sum of squares error functions gives

$$E = \frac{1}{2} \sum_n \sum_i (y_i(\mathbf{x}^n) - t_i^n)^2, \tag{8.4}$$

where n sums over all the training samples and t_i^n is the target output at node i of sample n. Substituting (8.3) and minimising (8.4) leads to

$$\mathbf{K^T}\ \mathbf{K}\ \mathbf{W^T} = \mathbf{K^T}\ \mathbf{T}, \tag{8.5}$$

where $\mathbf{K}$ is the kernel matrix with on the rows the hidden layer outputs of sample n (i.e. elements $k_i(\mathbf{x}^n, \mu)$), and $\mathbf{T}$ the target matrix that has the targets of sample n as rows (i.e. elements t_i^n). The solution is then written as

$$\mathbf{W^T} = \mathbf{K}^\dagger \, \mathbf{T}, \tag{8.6}$$

with $\mathbf{K}^\dagger$ the pseudo-inverse of $\mathbf{K}$ defined as $\mathbf{K}^\dagger \equiv (\mathbf{K^T} \, \mathbf{K})^{-1}\mathbf{K^T}$.

8.3 RBF Classification

To minimise the number of tables and results, the classification will focus immediately on the whole data set instead of the five sets used earlier. Classification with RBF was a significant improvement over the results shown in Chapters 6 and 7 and it is not necessary to first demonstrate it on smaller subsets. The data was prepared and features were extracted as described in Chapter 7, i.e. the full bandwidth was evaluated for feature selection and following the same reasoning the data was low-pass filtered, lightly denoised, down-sampled, and final features were selected from the local discriminant basis. The selected features are shown in Table 8.1. The selected features are very similar to the optimal features found for the smaller data set in Table 7.3. The main difference is that the features are even more focussed on the lower frequencies and the questionable feature (as it fell outside the filter's pass-band) in the frequency band 2250 - 3000 Hz was not selected. The fact that similar features were selected could indicate that these features are globally discriminating and could be used directly for other data sets, but this needs to be evaluated. Especially considering that for female animals there might be more focus on frequencies around 1500 Hz.

The RBF classification approach has a number of parameters that are considered to be fixed while the neural network is trained, and parameters that are optimised during training. The parameters that are considered fixed are the following :

- number of features
- feature selection itself (i.e. location in the wavelet packet table)
- number of hidden layer nodes
- standard deviation of Gaussian kernel

These parameters should not change too much between different data sets. The number of features already more-or-less cover the low frequency bandwidth (11 of the available 16 features in the 1 - 750 Hz bin were selected), perhaps the number could be reduced but there is not much room for increment. The feature selection itself showed to be stable when the number of classes was increased. The number of hidden layer nodes will be evaluated below, it is difficult to predict the number of clusters that will be necessary in general to model the fairly sparse feature space. As clustering is an expensive operation, it would be preferred to find a reasonable number of clusters that can always be used, perhaps erring on the side of lightly over-fitting the data. Avoiding the need to re-cluster data in real-time to find an optimal number of clusters will lower the required CPU cycles. In Section 8.4 another classifier will be discussed that does find an optimal number of clusters that allows to separate two classes. The standard deviation is considered not to be a critical parameter on the performance of the neural network. It will be investigated below, but in general it should be large enough to create overlap between the clusters, to cover the feature space sufficiently to allow generalisation. The values of the selected wavelet coefficients should not vary too much in range, as the patterns are always normalised in energy, meaning that the features are expected to always

have the same order of magnitude.

Parameters that are optimised during training are :

- cluster centre means to model the feature space
- network weights

The following sections will first focus on the clustering and on obtaining classification results with reasonable values for the fixed parameters. Then variation in these parameters is investigated to see how the results are influenced.

index	split	frq band (Hz)	power	index	split	frq band (Hz)	power
1	3	1 - 750	3.2	2	3	1 - 750	1.5
3	3	1 - 750	4.9	4	3	1 - 750	5.8
5	3	1 - 750	2.4	6	3	1 - 750	0.74
7	3	1 - 750	0.67	8	3	1 - 750	1.3
9	3	1 - 750	1.5	10	3	1 - 750	1.0
11	3	1 - 750	2.1	21	3	750 - 1500	0.68
22	3	750 - 1500	0.71	23	3	750 - 1500	1.9
24	3	750 - 1500	1.0			-	

Table 8.1: Selected wavelet packet coefficients for discrimination of the full data set.

8.3.1 Clustering of features

The features were first clustered with the adapted *c*-means clustering algorithm described in Section 5.2. This algorithm itself has a parameter, the critical value of the Anderson-Darling test. Of course, the interest here is not to find truly normally distributed clusters. Therefore the critical value does not have to be set at a specific level. A high critical value (and low p-value) will accept cluster more easily, or be less strict on the cluster shape, limiting the number of clusters found (the term 'accept' is not used here in the statistical sense that the H0 hypothesis would be accepted). On the other hand, a low critical value will reject many clusters forcing the algorithm to split up the data into many smaller clusters. The critical value becomes a parameters that can be used to steer the algorithm towards a specific number of clusters, but a questionable assumption here is that the clusters should be normally distributed in the direction of their principal component. The plots in Figure 8.1 do not immediately validate this assumption, and while testing it was found that *c*-means generally underestimated the number of clusters needed for optimal performance. As such, it was used to get an indication of the number needed.

There are two ways the feature space can be clustered, one approach would be to combine all data together, another one is to model the classes individually. The former algorithm has the benefit of needing less clusters to cover the feature space, for example when clusters overlap. However, the number of total clusters is small anyway, and as mentioned we preferred to err on the side of lightly overfitting the data than underfitting. This would allow to fix the number of clusters in general for clustering (including on other data sets in real-time embedded situations) and give enough flexibility in cases where classes would not overlap or the feature space would require a centre more to be modelled. The preference of overfitting came from testing with *c*-means and minimising the number of centres which consistently resulted in worse performance than clustering with a slightly larger number.

Clustering the whole feature space together with c-means using a critical value of 0.87 (corresponding to a 2.5% level test) typically resulted in ca. 7 clusters for combinations of 4 classes and 14 when all classes are used. Using a cv of 0.63 (corresponding to a 10% level test) these values respectively changed to 10 and 21. Individual clustering of the classes resulted in 17 (cv = 0.87) and 20 (cv = 0.63) clusters. Based on these results, it seems that the individual classes can be modelled using 2 or 3 clusters. It also seems that there was not a large difference in the number of clusters whether clustering is done directly over all classes, or individual classes. This indicates good separation in the feature space. While the outcomes were comparable, we still preferred to cluster individual classes to obtain more accurate models, including for cases where there might be overlap.

8.3.2 Data classification

The seven available classes were classified with the RBF network, using the features from Table 8.1 and modelling the feature space with 3 clusters per class. In the end, after k-means clustering, one cluster that had less than 5 points was discarded leading to 20 clusters in total. The standard deviation of the Gaussian kernels was kept fixed at 0.3; the results are shown in Table 8.2. Performance on the training data was already much better than when using the simpler linear algorithm that was used for Table 7.4. But the important improvement is in the validation set, where all seven animals have reasonable classification. As noted in the description of the data in Appendix A, the last validation set labelled as 7* contains all clicks from the dive of the animal that were not in the single click train used for training (where the set labelled 7 contains the clicks from the same click train). In the following these results will be used as the reference to compare possible improvements to. The interest is especially in improving performance on the second class, without deteriorating the other classes too much. Based on the classification of the second class it could be argued that the performance on the training set is too good and that the classifier has been overfitted.

	Training set							Validation set							
Classified as	1	2	3	4	5	6	7	1	2	3	4	5	6	7	7*
Set 1	100	0	0	0	0	0	0	84	7	6	3	0	0	0	6
Set 2	0	100	0	0	0	0	0	4	67	0	0	0	0	2	8
Set 3	0	0	100	0	0	0	2	10	1	91	1	1	2	1	4
Set 4	0	0	0	98	0	0	0	0	0	0	92	3	0	0	2
Set 5	0	0	0	2	100	0	0	1	7	3	4	96	0	1	2
Set 6	0	0	0	0	0	100	0	1	11	0	0	0	78	1	8
Set 7	0	0	0	0	0	0	98	1	7	0	0	0	20	95	70

Table 8.2: Click classification using a radial basis function network with 15 features and 4 clusters per class.

8.3.3 RBF parameter selection

Good values for parameters are often found through a brute-force search. Trying out all possible combinations would be very computationally expensive, therefore the approach used is to fix all but one parameter, which is then varied to find optimal performance for the classifier and the process is then repeated with another parameter. This does not guarantee an optimal combination of parameters in the end, but it should give a good enough solution.

The number of clusters per class already has been investigated in Section 8.3.1 and the position of the features (their selection from the wavelet packet table) will be considered fixed. However, the number of features and kernel standard deviation can be varied to attempt to lower a possible bias of the classifier to the training data. To evaluate the classifier with changing parameters the training data will be organised

somewhat differently. Testing classification against the remainder of the click trains will certainly bias the parameters to this specific data set and might reduce performance on other data. Another point is that the classifier is supposed to be trained using only the first minute of data, and if parameters need to be set in real-time, it will have to be done based on this available minute. To this end, selection of parameters was done using a bootstrapping technique [34]; the method used here differs from the publication in [95], per article reviewer request, but the obtained results were comparable. A training set was created taking uniformly distributed random samples with replacement from the original training set (first 50 clicks). The classifier was then trained with this set and validated with samples from the original set that were not included for training. This process was repeated several times for each configuration of the classifier.

Weight decay regularisation

One method that is often used to improve generalisation is weight decay regularisation. This adds an error as a function of the weights to the optimisation function (8.4), usually in the form of

$$E(\mathbf{w}) = \lambda \|\mathbf{w}\|^2. \tag{8.7}$$

This forces weights that do not have much influence on the classification itself to be small, and allows them to be pruned later to improve generalisation. In the case presented here, the number of weights (i.e. the number of kernels in the hidden layer) is decided by a clustering algorithm and already quite small. In addition, after k-means clustering small clusters are already discarded reducing the complexity of the classifier. As the number of parameters is not expected to be a problem, and is already controlled, weight decay regularisation was not further considered. Instead, attention went to the number of features that are selected from the wavelet packet table and the width of the hidden layer kernels.

RBF cluster standard deviation

The role of the standard deviation in the kernel is to create a localised classifier while at the same time allowing generalisation (requiring some spread in the feature space). It is not necessary that it accurately describes the shape of the data cluster that is being modelled by the kernel, nonetheless classification was tested using diagonal and full covariance matrices taken from the data which did not improve results. To select a good value, a range was tested between 0.05 up to 0.5. Fifty bootstrap sets with 40 patterns of each class were created for each value tested. This gave corresponding validation sets of at least 10 patterns per class, but usually much higher due to sampling with replacement. Each classification created a result matrix, which were summed over all the available bootstrap sets.

To evaluate the performance of the classifier, true classification and false classification rates, similar to true and false positive rates that are used in repeater operating curves, were computed by summing respectively the diagonal and off-diagonal results of the classification matrix. These were divided by the total number of positives (total number of patterns in all validation sets) and the total number of negatives (all validation patterns that did not belong to a specific class). An example is given in Table 8.3 which shows the cumulative classification of 50 bootstrap sets with $\sigma = 0.1$. The true classification rate here would be the diagonal sum divided by total samples, or $7742/7980 = 0.97$. The false classification rate would be the off-diagonal sum divided by total negatives, or $238/6840 = 0.035$.

Varying the standard deviation and following the bootstrap process led to the results in Table 8.4, which used 100 bootstrapped sets for each σ. Based on these results, a value around 0.20 and higher should give optimal generalisation. The classifier is not very sensitive to an exact value of σ (according to this training set), which could be expected. Once the kernels are broad enough to overlap and cover the feature space, they can generalise to the validation set. When they grow beyond that it takes 'a while' before they are so broad that they start losing their capacity to differentiate. It is not always clear to pick a 'best' value,

	Bootstrap validation set						
Classified as	1	2	3	4	5	6	7
Set 1	1133	0	0	0	0	8	0
Set 2	0	1111	0	7	0	10	6
Set 3	0	1	1128	0	0	1	0
Set 4	0	2	2	1069	0	6	10
Set 5	0	0	0	10	1139	0	0
Set 6	0	0	0	0	0	1039	1
Set 7	7	26	10	54	1	76	1123

Table 8.3: The cumulative classification matrix of 50 bootstrap sets.

often there is a trade-off between one class performing better at the expense of another class. The complete classification when using $\sigma = 0.20$ and $\sigma = 0.40$ is shown in Table 8.5. Comparing to Table 8.2, there has been improvement in the generalisation of Set 7, including to the complete dive, but at the cost of especially Set 6 when $\sigma = 0.20$ and Set 2 when $\sigma = 0.40$. Possible overfitting of Set 2 has clearly not been resolved and a σ of around 0.30 is a reasonable choice.

σ	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
TCR	81.9	97.1	99.0	99.0	99.0	98.8	98.9	98.8	98.8	98.7
FCR	21	3.3	1.2	1.2	1.2	1.5	1.3	1.3	1.4	1.5

Table 8.4: Classifier performance varying the value of the standard deviation, measured in true classification (TPR) and false classification rates (FPR).

	$\sigma = 0.20$								$\sigma = 0.40$							
Classified as	1	2	3	4	5	6	7	7*	1	2	3	4	5	6	7	7*
Set 1	85	7	18	3	0	0	0	4	84	7	6	3	0	0	0	6
Set 2	6	66	0	0	0	0	1	7	3	64	0	0	0	0	1	8
Set 3	8	2	79	1	1	0	1	3	11	2	90	1	1	4	1	3
Set 4	0	0	0	92	3	0	0	2	0	0	0	93	3	0	0	1
Set 5	0	4	3	3	96	0	0	1	2	11	3	4	96	0	1	2
Set 6	1	17	0	1	0	63	1	9	1	8	0	0	0	80	1	8
Set 7	1	4	0	1	0	37	98	75	1	7	0	0	0	17	96	71

Table 8.5: Click classification using radial basis functions with $\sigma = 0.20$ and $\sigma = 0.40$, otherwise parameters are as in Table 8.2. Only the results of validation are shown.

Number of features

The 15 features from Table 8.1 gave good performance although they may overfit the data. Therefore, this number is considered to be a maximum, and the question is if there could be a smaller number that will improve the generalisation.

First the weak features were removed to assess their importance to the classifier. The training results are shown in Table 8.6. The table shows an optimal value when no dimensions are discarded. This could be

a reason to search for more features, but these were not added out of fear for overfitting the training data set further. Complete classification for both the 14 and 10 strongest features is shown in Table 8.7. Again, performance on the second class has not been improved, in fact, most classes performed worse for either value. In an embedded real-time setting, if there were pressing reasons to reduce the number of features for memory or processing reasons, the fairly constant behaviour between 9 and 12 features would make a value from this interval a good candidate.

# features	15	14	13	12	11	10	9	8	7	6	5	4
TCR	98.8	98.8	97.8	97.2	97.2	97.4	97.6	95.7	95.1	93.5	91.0	88.9
FCR	1.4	1.4	2.6	3.2	3.2	3.1	2.8	5.0	5.7	7.6	10	13

Table 8.6: Classifier performance when the number of features are reduced in order of weakest power of discrimination.

	14 features								10 features							
Classified as	1	2	3	4	5	6	7	7*	1	2	3	4	5	6	7	7*
Set 1	75	7	6	3	0	0	0	6	77	10	3	3	0	0	0	5
Set 2	4	69	0	0	0	0	1	8	8	59	15	1	0	0	2	7
Set 3	19	1	91	1	1	2	1	5	12	0	79	1	0	2	1	2
Set 4	0	0	0	91	3	0	0	2	1	0	0	80	4	7	0	1
Set 5	1	7	3	5	96	0	1	2	0	1	3	15	96	0	1	3
Set 6	1	9	0	0	0	85	1	9	2	28	0	1	0	76	6	18
Set 7	1	7	0	1	0	13	96	67	1	2	0	0	0	15	89	63

Table 8.7: Click classification using radial basis functions ($\sigma = 0.30$) on the original validation sets with a reduced number of features. Only the validation sets are shown.

The next question is if there is strong correlation between the features. The (upper-half) of the correlation matrix is shown in Table 8.8. Some high values can be found, with an absolute maximum of 0.72 between the third and fourth feature, but generally they are not so high to immediately suggest a candidate dimension for removal. Notwithstanding, highly correlated dimensions were removed and classification was tested, with similar results as were reached by removing weak dimensions. That is, generally the (bootstrapped) testing and validation data sets classified very good, with the same poor generalisation to the second set and overall worse performance for the complete dive.

Instead of simply removing dimensions, there are many projections that allow dimension reduction (e.g. principal components, Fisher's linear discriminant). Still, the objective is not to encode the same information in less dimensions, but to reduce the information to improve generalisation. There is no obvious choice for a projection and based on the results in the last sections, where training and validation always went perfectly together with poor generalisation to the left over test set, the problem may be that the training data taken from the start of a click train is simply not representative enough to cover later data, especially in the case of the second set.

8.3.4 A closer look at Set 2

To understand why the second set behaved so poor, and to see e.g. if problems were caused by burst errors or random errors, the outputs were plotted in a graph. In Figure 8.3, the left graph are the outputs of Set 2, on the right the outputs of Set 4. The green dots are outputs from the correct (class) output nodes, while

R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.0	-0.37	-0.64	0.56	0.49	-0.41	-0.21	-0.29	-0.25	-0.32	0.61	0.62	-0.22	0.45	-0.12
2		1.0	-0.20	-0.17	-0.19	0.082	0.015	0.21	0.0046	0.21	-0.21	-0.15	-0.12	-0.085	0.17
3			1.0	-0.72	-0.43	0.34	0.28	0.043	0.18	0.37	-0.42	-0.46	0.31	-0.55	0.19
4				1.0	0.020	-0.30	-0.53	0.054	0.14	-0.40	0.32	0.37	-0.36	0.66	-0.43
5					1.0	-0.53	0.15	-0.23	-0.49	-0.23	0.34	0.38	0.096	0.051	0.29
6						1.0	0.11	-0.15	0.13	0.10	-0.25	-0.31	0.12	-0.16	-0.26
7							1.0	-0.42	-0.43	0.093	0.043	-0.030	0.37	-0.38	0.32
8								1.0	0.29	-0.22	-0.50	-0.17	-0.27	-0.042	-0.072
9									1.0	-0.18	-0.58	-0.23	-0.19	0.000	-0.16
10										1.0	-0.073	-0.27	0.040	-0.30	0.26
11											1.0	0.42	0.12	0.40	-0.11
12												1.0	-0.055	0.26	0.031
13													1.0	-0.068	-0.067
14														1.0	-0.61
15															1.0

Table 8.8: Correlation between the 15 selected features from the local discriminant basis.

the red dots are outputs at the remaining nodes. The first 45 samples were results from training data. To smooth the curves a 5-point moving average was used (first few samples were skipped as they were averaged with zeroes outside the window). Both sets have outputs around 1 for their training samples, the target values with which they were trained. As was typical for all data sets, after training the correct output value drops fairly fast, while the wrong outputs stay reasonably close around 0. However, in the case of Set 2, there seems to be a downward trend that does not reverse and the samples are no longer easily distinguishable from the other classes. In contrast, the correct output of Set 4 also lowers, but it does not follow a continuously lowering trend and stabilises around an output of 0.6. It does reveal a burst error with one other class that shows ascending outputs.

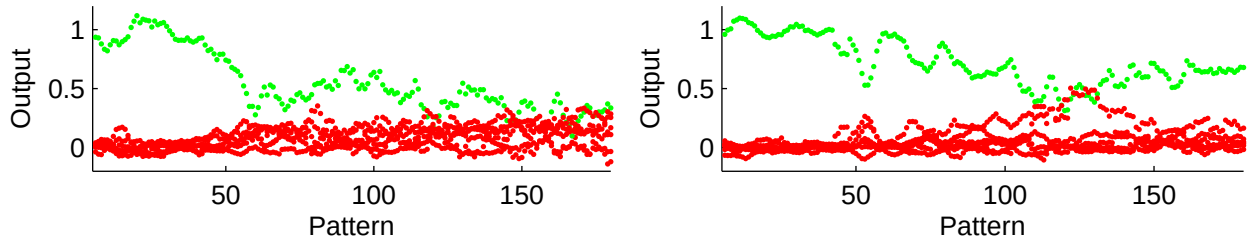

Figure 8.3: Network outputs of Set 2 (left) and Set 4 (right). The green dots are the outputs given at the correct output node, the red dots are outputs at the other 6 nodes. The first 45 samples are outputs from the training sets; the lines are smoothed with a 5 point moving average.

One possible explanation could be a poor signal to noise ratio in the recording of the second animal. To assess this the left graph in Figure 8.4 shows the signal to noise ratios around the clicks in Sets 2 (green) and 4 (red). The SNR was computed taking 10 ms of the signal and 1 s of the noise just before the signal, not paying attention to any echo or other signal that might accidentally come before it and after filtering the data between 100 and 3000 Hz to reflect the noise in the frequency bands the classification is performed in. Basically the graph shows the proportion that the click sticks out above the noise. It can be seen that

Set 2 follows a trend towards 0 dB, where Set 4 remains roughly above 10 dB. The SNR seems to follow somewhat the trends of the correct classifier outputs in Figure 8.3, but that is speculative.

Another cause for problems is displayed on the right side of Figure 8.4. Here, two clicks are superimposed from the start of the click train in blue and the end of the train in red, both were filtered to be in a 100 - 5000 Hz frequency band. They were synchronised on the slope of the first minimum around sample 120. However, the blue click has some interference, visible at the start through the clipping of a peak, that causes the remainder of the two patterns to be unsynchronised and even phase inverted at some points, e.g. around samples 160, 220, 260. This could be due to noise, or effects from the animal's orientation as was explained in Section 6.3. Either way, it will lead to considerable variance in the wavelet coefficients, which, when not accounted for in the training set, will position kernel clusters incorrectly (at least to allow generalisation).

Difficulties in generalisation due to a non-representative training set can be demonstrated by not using only the first 50 clicks, but by creating a training set based on 50 randomly selected clicks from the available click train. Classifying with the default settings also used for Table 8.2 led to the results in Table 8.9. Two things can be noticed when comparing the two tables, first the training data no longer gives close to perfect classification, which might indicate less bias. And second, generalisation was greatly improved. Especially the performance on the second set increased, which now was classified correctly for 88% of the patterns, but in fact all sets showed improvement. The complete dive (Set 7*) is still at 71%, however it should be kept in mind that only (random) data from a single click train was used for training, thus in this case the training data may still not represent the whole data set.

It will not be practical to use so much information for training when the algorithm is used in the field, but these results do show that there are no inherent problems for the classifier to perform its task. It also motivates to see if there still is some way to improve generalisation using only the first 50 clicks.

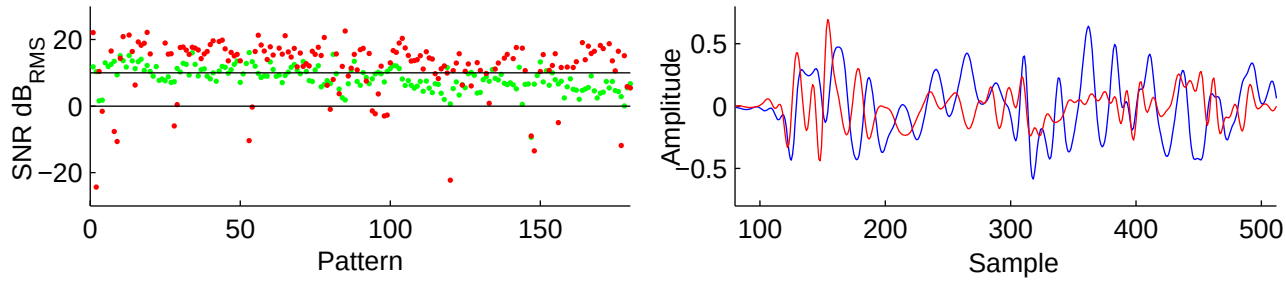

Figure 8.4: Reasons for poor classification, on the left image the signal to noise ratio is plotted in dB_{RMS}. Red dots are the local SNR at the clicks in Set 4, green dots in Set 2. The right image shows two superimposed patterns from Set 2, from the start of the click train (blue) and from the end (red).

8.4 Support Vector Machines

Classification with a radial basis function network has been shown to be reasonably effective. The training approach was based on two separate steps of first modelling the feature space and then finding optimal output weights. The number of clusters used in the hidden layer was decided by the clustering algorithm and were kept fixed while finding optimal classification results. The main problem with the classification was that the training set did not completely represent the data and the RBF network did not always generalise well. Interestingly, there is another classifier that uses the exact same schematic as RBF (Figure 8.2), but

	Training							Validation							
Classified as	1	2	3	4	5	6	7	1	2	3	4	5	6	7	7*
Set 1	96	0	0	4	0	0	0	98	3	0	1	1	0	0	4
Set 2	4	96	0	0	2	0	0	2	88	0	0	1	0	1	11
Set 3	0	0	98	0	0	0	0	0	1	97	1	0	0	0	2
Set 4	0	0	0	94	0	0	0	0	0	0	95	3	0	0	3
Set 5	0	2	2	2	98	0	0	0	1	3	3	94	0	0	2
Set 6	0	0	0	0	0	100	0	0	2	0	0	0	100	1	8
Set 7	0	0	0	0	0	0	100	0	5	0	0	0	0	99	71

Table 8.9: Click classification using radial basis functions with $\sigma = 0.30$ and a training set composed of 50 randomly selected clicks from the click train.

combines training of the two layers and finds an optimal number of clusters, known as support vector machines. This type of algorithm is known for its capacity to generalise (e.g. [90]), although it is more complex than a RBF network.

The difference between the two network approaches is in the underlying model. Where the presented RBF network models the centres of highest density in the data and can classify multiple classes at the same time, SVM solves the 2-class classification problem by first using a non-linear mapping to project the data to a high dimensional feature space, where the classes are then separated linearly. It does this by maximising the boundary between classes and as such may be less influenced by 'strange' shapes in the (projected) distribution of the data. A draw back (or perhaps advantage) of SVM here is that a single machine will only distinguish between two classes. This means that in situations with more than two classes, multiple machines are necessary. There are generally two approaches to handle the multi-class case, either machines are trained 1-vs-all, needing N machines for N classes, or a single machine is used for any two pairs of classes, needing N*(N-1) machines for N classes. The advantage may lie in the fact that a single machine can be very specifically trained to recognise a single class. It is possible that this focus on a class boundary rather than class centres will improve the classification of the sperm whales where the class centres may undergo some change throughout a dive.

There are generally two different algorithms used to train a support vector machine known as C-SVM and ν-SVM. They differ in the way errors are penalised, where ν-SVM allows better control over both the training error and the number of support vectors that are used (theoretically both training methods can lead to the exact same classification results).

8.4.1 SVM description

The architecture of a SVM is basically identical to the RBF architecture shown in Figure 8.2, but it differs in the underlying approach and the way the network is trained. Restricting, for the moment, to classification between two classes, SVM tries to find a hyper-plane that optimally separates these two classes, i.e. a hyper-plane such that

$$\begin{aligned} \mathbf{w}^T\mathbf{x}_i + b &\geq 1 \qquad \text{when} \quad t_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b &\leq -1 \qquad \text{when} \quad t_i = -1 \end{aligned} \tag{8.8}$$

where $\mathbf{w}$ is the direction of the normal vector on the separating plane, b the bias, and $\mathbf{w}$ and b have been scaled to achieve the normalised bounds on the right-hand side. A hyper-plane can then be considered

optimal when the distance between two points belonging to different classes on either side of the plane is maximised. This maximum margin is achieved at the minimum $||\mathbf{w}||$. However, considering outliers and possibly cases where the data are not linearly separable, the maximum margin might be 0. Therefore, it can be preferred to relax the requirement that all data should be outside the margin as follows :

$$t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \tag{8.9}$$

with $\xi_i \geq 0$. This allows a number of the $\mathbf{x}_i$ to be inside the margin (or even wrongly classified). The optimal hyper-plane is then found by minimising both $||\mathbf{w}||$ and $\sum_{i=1}^{N} \xi_i$. In order to control the error rate and margin crossings, the ξ's are multiplied by a penalty C; the resulting minimisation problem thus becomes :

$$\min_{\mathbf{w},\xi,b} \; ||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i \tag{8.10}$$

under the constraints

$$t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \tag{8.11}$$

$$\xi_i \geq 0. \tag{8.12}$$

Note that for a large value of C, crossing the margin will be penalised more, while small values allow more points to cross the margin. The minimisation problem (8.10) has a straightforward solution using Lagrange multipliers. The details can be found in [24]; the optimal $\mathbf{w}_o$ can be expressed as follows

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i t_i \mathbf{x}_i \tag{8.13}$$

where α_i are the Lagrange multipliers, which are found by solving

$$\max_{\alpha} \; \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i\alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{8.14}$$

under the constraints

$$0 \leq \alpha_i \leq C, \quad i = 1, .., N; \quad \sum_{i=1}^{N} \alpha_i t_i = 0. \tag{8.15}$$

Normally, only a few of the α_i's will be non-zero, the $\mathbf{x}_i$'s related to these non-zero values are called the support vectors and are the only ones that are decisive for the separating plane, all others can be discarded. These support vectors lie either on the margin, or exceed it. Finally, b_o can be calculated using the support vectors lying on the margins (which have $\xi_i = 0$), and the classification of an unknown pattern $\mathbf{x}$ is then done based on the sign of the decision function

$$f(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o. \tag{8.16}$$

This approach to handle non-separable classes is called C-SVM, another method is known as ν-SVM [22, 90]. The latter variant allows more direct control of the margin, in that case defined by

$$\mathbf{w}^T\mathbf{x} + b = \pm\rho, \tag{8.17}$$

with ρ left as a free variable that can be optimised. The minimisation problem now becomes

$$\min_{\mathbf{w},\xi,b,\rho} \ \frac{1}{2}||\mathbf{w}||^2 - \nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i \tag{8.18}$$

under the constraints

$$\begin{gathered} t_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho - \xi_i \\ \xi_i \geq 0; \quad \rho \geq 0. \end{gathered} \tag{8.19}$$

The solution for optimal $\mathbf{w}_o$ is given by :

$$\mathbf{w}_o = \sum_{i=1}^{N}\alpha_i t_i \mathbf{x}_i \tag{8.20}$$

where α_i are found by solving

$$\max_{\alpha} \ -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{8.21}$$

under the constraints

$$0 \leq \alpha_i \leq \frac{1}{N}; \quad i = 1, .., N; \quad \sum_{i=1}^{N}\alpha_i t_i = 0; \quad \sum_{i=1}^{N}\alpha_i \geq \nu. \tag{8.22}$$

This second formulation gives control through the constant ν over both the error rate on the training set and the number of support vectors from the optimisation through the relationships [90]:

$$P_e \leq \nu; \qquad \text{and} \qquad N\nu \leq N_s. \tag{8.23}$$

This can be a useful property as the number of support vectors play a large role on the speed of the network.

The classification as described above is performed in the same space as the input patterns, and limited to a linear classifier. Since relationships between characteristics can often be non-linear, it can be advantageous to map them into another feature space that linearises these relations. The drawback is that this mapping may also increase the number of dimensions by several orders of magnitude and in many cases this becomes intractable. Looking again at (8.16), expanding $\mathbf{w}_o$ and mapping the input patterns $\mathbf{x}$ into a feature space $\mathbf{x} \mapsto \phi(\mathbf{x})$, leads to the following

$$f(\mathbf{x}) = \sum_{i \in \text{ sv}}\alpha_i t_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b_o, \tag{8.24}$$

where sv denotes the restriction to the support vectors. Instead of physically mapping the points $\mathbf{x}_i$ and $\mathbf{x}$ into the feature space, if ϕ maps the features into a Hilbert space, it is also possible to calculate the product directly using a kernel function, defined as

$$k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y}). \tag{8.25}$$

When a kernel function is constructed that is significantly easier to evaluate than the mapping ϕ, the higher dimensionality is no longer time prohibitive (other high dimensionality issues that normally affect the performance of a classifier are covered in detail in [90], or most other SVM texts). In the case of minimisation problem ((8.14) or (8.21)), it is only necessary to evaluate the dot product of the input vectors $\mathbf{x}$, and an explicit mapping of $\phi(\mathbf{x})$ is never required ($\mathbf{w}_o$ is never evaluated). Using a kernel function (8.24) becomes

$$f(\mathbf{x}) = \sum_{i \in \text{sv}} \alpha_i t_i k(\mathbf{x}_i, \mathbf{x}) + b_o. \tag{8.26}$$

Again, the Gaussian kernel given in (8.2) is often used.

The SVM only separates two classes, but the same algorithm can easily be extended to several classes [8], for example by using a one-against-one approach. For this classification method a separate machine is created for every combination of two classes, meaning that when there are n classes this results in $\frac{n(n-1)}{2}$ machines. A new pattern is then presented to all the machines, and the class that occurs most frequently in the outcome is chosen. In the case when there are two or more classes with identical frequencies, the pattern is defined as unclassifiable.

8.5 SVM Classification

The SVM classifier has almost the same parameters as those that were used with RBF classification, parameters that are considered fixed during training are the following :

- number of features
- feature selection
- kernel standard deviation
- regularisation parameter

While multiple SV machines will be used for handling multiple classes, where in principle each machine could have its own set of parameters, they will all share the same values to limit the number of parameters that need to be tuned. The feature selection itself will be considered fixed again as was done with the RBF network. Additionally, the number of features will also be kept at 15. For RBF networks the performance with less features deteriorated quite quickly when the whole data set was considered, while performance was always very good on the bootstrapped training data. As shown below, the training data still poorly represented the whole data set for the SVM classifier and with generalisation in mind the number was not changed.

Parameters that are optimised during training are :

- number of kernels
- kernel centres
- output weights

A change with RBF is that the number of kernel centres has now become a parameter that is optimised. At the same time, their centres are not really optimised by themselves. The kernels are defined by the support vectors, i.e. data points that lie on or within the margin, which basically fixes the possible positions of the kernels.

An initial classification was made with C-SVM, using $\sigma = 0.3$, $C = 0.50$ and a one-against-one machine strategy, with results in Table 8.10. Comparing with Table 8.2, generalisation to the complete dive, Set 2 and especially the sixth set had been improved, while on the other hand Sets three and four classified worse. The percentage of undecided patterns is reasonably low. The total number of support vectors used by the classifier was 337, an average of 16 per machine with 21 machines in total. This is fairly high considering that the RBF classifier only needed around 20 in total. It might be an indication of overfitting and to compete with RBF the number needs to be brought down using the regularisation parameter in either C-SVM or ν-SVM.

	Training							Validation							
Classified as	1	2	3	4	5	6	7	1	2	3	4	5	6	7	7*
Set 1	100	0	0	0	0	0	0	85	5	13	3	0	0	0	4
Set 2	0	100	0	0	0	0	0	8	70	0	0	0	0	1	6
Set 3	0	0	100	0	0	0	0	1	0	81	1	0	0	0	2
Set 4	0	0	0	98	0	0	0	0	0	0	80	1	0	0	1
Set 5	0	0	0	2	100	0	0	0	2	3	15	99	0	0	3
Set 6	0	0	0	0	0	100	0	1	13	3	0	0	100	1	9
Set 7	0	0	0	0	0	0	100	5	9	0	1	0	0	98	75
undecided	0	0	0	0	0	0	0	2	5	3	0	1	0	0	3

Table 8.10: Classification using C-SVM with $\sigma = 0.30$ and $C = 0.5$. A total of 337 support vectors were used by the classifier.

8.5.1 SVM parameter selection

To find suitable values for the parameters the same protocol was followed as was described in Section 8.3.3, using bootstrapping to create 100 training and corresponding validation sets from the first 50 clicks of the click trains. The results were then summarised again in tables describing the false and true classification rates, where undecided patterns were added to the false classifications. Additionally, the number of support vectors was stored for each classification result to keep track of the amount of information that was used; the tables will show the average number of support vectors that were used per machine, there were always 21 machines in total.

Regularisation and kernel standard deviation parameters for C-SVM

First, the false classification penalty for C-SVM was looked at, using $\sigma = 0.30$ as initial value for the kernel width, which was a reasonable value for the RBF network (although the kernels were positioned differently there). The classification results are shown at the top in Table 8.11. The classification converged fairly fast to an optimal solution for $C \geq 0.9$. The fact that the performance remained more or less equal while strongly discouraging wrong classifications during training with increasing values of C, indicates that the training sets were fairly well separated by their boundary elements and that there were very few misclassifications. The validation set was perhaps too similar to the training set to allow later generalisation to the entire click train. The close to perfect separation did require roughly 6 times more centres than were needed for the

RBF network.

Next, a value for σ was sought while keeping C fixed at 1 based on the top of Table 8.11. Although the table might suggest higher values, it has to be kept in mind that the validation set based on the first 50 clicks is quite limited and not representative for all data, therefore it was decided not to completely disallow wrong classifications. Varying σ led to the results at the bottom in Table 8.11. Here, the effects on the classification were more apparent. A small σ did not cover the feature space appropriately and required a large number of support vectors. Increasing it gave the region $0.27 \leq \sigma \leq 0.33$ with reasonable performance versus the number of support vectors.

C	0.1	0.5	0.9	1.3	1.7	2.1	3	4	5
TCR	96.3	98.9	99.2	99.2	99.3	99.2	99.5	99.6	99.2
FCR	4.3	1.3	0.92	0.95	0.83	0.92	0.64	0.52	0.97
# SV	58	25	20	18	17	16	16	15	15
σ	0.15	0.20	0.25	0.27	0.29	0.31	0.33	0.35	0.40
TCR	99.4	99.4	99.2	99.2	99.2	99.1	99.1	98.9	98.8
FCR	0.71	0.72	0.93	0.89	0.92	1.1	1.0	1.3	1.4
# SV	32	24	20	20	19	19	19	19	21

Table 8.11: Top: Evaluation of the error penalty in C-SVM while $\sigma = 0.30$. Bottom: Evaluation of the kernel width in C-SVM while $C = 1$. The number of support vectors is the average per machine (21 in total).

The number of support vector machines in Table 8.11 was quite high, which is why training using ν-SVM was also considered. Results for a range of values for ν and σ are shown in Table 8.12. It can be seen that for increasing ν, or increasing the margin, the bounds in equation (8.23) are also increased, allowing more errors and increasing the lower bound on the number of support vectors. From this table, a reasonable value for ν would be between 0.11 and 0.17 in order to limit the number of support vectors. A corresponding σ could be any value around 0.30.

In all resulting SVM tables shown so far the effect of the parameters seems quite small. This is likely due to the training and validation sets being very similar and the data set used for training apparently separates quite well when SVM are used. It makes it difficult to find appropriate values that are expected to work well on all available data. A better demonstration of the parameter's influence is shown in Figure 8.5. Here, the classification of the original validation set of the Sets 2 (solid) and 7* (dash-dotted) is shown while varying σ, in this case ν was set to 0.15. Both these sets showed the poorest generalisation of all available sets, and of course the complete dive is most interesting to compare to. Additionally, the total number of support vectors used by the SVM is shown with the dotted line and indexed on the right axis. As can be expected, a large standard deviation decreases the number of support vectors, as each vector covers a larger area. At the same time, generalisation to the complete dive initially improves somewhat before deteriorating after $\sigma = 0.3$. As with RBF, Set 2 always seems problematic when training data is restricted to the first 50 clicks.

Classification of the complete data set using both C-SVM and ν-SVM and setting parameters to reasonable values taken from Tables 8.11 and 8.12 gave the results in Table 8.13. The σ for ν-SVM was selected fairly low to get good generalisation on Set 7* at the cost of more support vectors. Comparison with for example RBF results in Table 8.5 shows little difference. SVM had more difficulties with Sets 3 and 4, but was much better on Set 6. ν-SVM was set-up to generalise best and as expected does so for Set 7*. It has to be concluded that with comparable results, the SVM performed worse because it needed much more information and took more time to train.

ν	0.05	0.09	0.11	0.13	0.15	0.17	0.19	0.25	0.30
TCR	98.9	99.2	99.3	99.3	99.2	99.3	99.2	99.0	98.8
FCR	1.3	0.92	0.87	0.80	0.93	0.82	0.88	1.1	1.4
# SV	13	14	15	17	18	19	21	25	28
σ	0.20	0.24	0.28	0.30	0.32	0.34	0.38	4.5	0.60
TCR	99.4	99.1	99.0	99.2	99.1	99.0	98.8	98.9	98.6
FCR	0.65	1.0	1.1	0.95	1.0	1.2	1.4	1.3	1.6
# SV	18	17	16	15	15	15	15	14	14

Table 8.12: Evaluation of the kernel width in ν-SVM while $\nu = 0.13$. The number of support vectors is the average per machine.

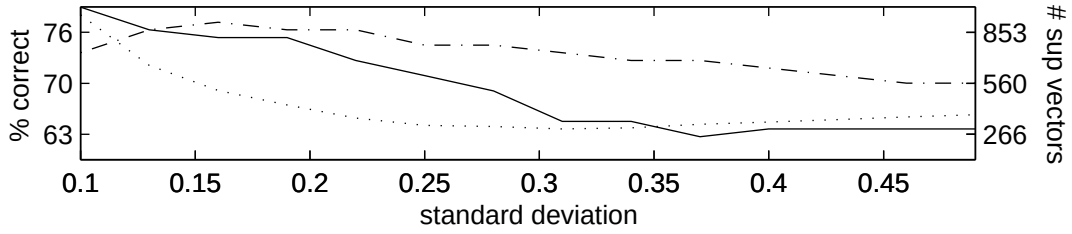

Figure 8.5: Percentages of correctly classified clicks of the original validation sets of Set 2 (solid) and Set 7* (dash-doted; long dive). The dotted line indicates the total number of support vectors that were used and is indexed on the right axis.

For completeness, the data was also classified with SVM using 50 randomly selected clicks from the click trains, shown in Table 8.14 for comparison with Table 8.9. It is not surprising that performance was much better than when using the 50 first clicks in Table 8.13, but for the individual click trains SVM also performed better than RBF. What is surprising is that the complete dive performed much worse than with RBF. The number of support vectors was quite high, possibly the data was somewhat overfitted. Alternatively, it is possible that for generalisation the classes were better described by their centres than by their (projected) boundaries. As the values can change during the dive, linear boundaries after the projection to the feature space may become invalid, while the distance to the cluster centres does not change too much.

	C-SVM, $C = 1.1$, $\sigma = 0.30$								ν-SVM, $\nu = 0.11$, $\sigma = 0.24$							
Classified as	1	2	3	4	5	6	7	7*	1	2	3	4	5	6	7	7*
Set 1	83	5	12	3	0	0	0	4	75	6	12	2	0	0	0	3
Set 2	8	66	0	0	0	0	1	6	9	70	0	1	0	0	1	7
Set 3	1	0	79	1	0	0	0	2	2	0	76	1	1	0	0	2
Set 4	0	0	0	80	1	0	0	1	0	0	0	80	3	0	0	0
Set 5	0	2	3	15	97	0	0	3	0	1	3	15	96	0	0	1
Set 6	1	13	3	0	0	100	1	8	1	14	6	1	0	100	1	8
Set 7	5	9	0	1	0	0	98	73	8	8	0	1	0	0	98	79
undecided	2	5	3	0	1	0	0	3	4	1	0	0	0	0	0	1

Table 8.13: Click classification using the two training methods for support vector machines. The left table used C-SVM and required 337 support vectors in total. The right table used ν-SVM and required 396 support vectors.

	C-SVM, $C = 1.1$, $\sigma = 0.30$								ν-SVM, $\nu = 0.11$, $\sigma = 0.24$							
Classified as	1	2	3	4	5	6	7	7*	1	2	3	4	5	6	7	7*
Set 1	98	1	0	2	0	0	0	4	98	1	0	2	0	0	0	4
Set 2	1	98	0	1	0	0	2	20	0	99	0	1	0	0	2	20
Set 3	1	0	97	0	0	0	0	1	1	0	97	0	0	0	0	1
Set 4	0	0	0	96	0	0	0	1	0	0	0	96	0	0	0	0
Set 5	0	1	0	1	100	0	0	1	0	1	0	1	99	0	0	1
Set 6	0	0	0	0	0	100	0	6	0	0	0	0	0	100	1	5
Set 7	1	0	3	0	0	0	96	63	1	0	3	0	0	0	98	68
undecided	0	0	0	1	0	0	2	3	1	0	0	1	1	0	0	1

Table 8.14: Click classification with SVM trained with 50 randomly selected clicks from the available click trains. C-SVM required 460 support vectors in total and ν-SVM 445.

8.6 Conclusion

After investigation of the feature space, a radial basis function architecture was trained to identify the sperm whales. It was shown that RBF networks are capable of distinguishing the animals using the features selected from a local discriminant basis. The results were compared with a support vector machine classifier. The main interest was to see if the classes were better described by their boundaries or by their centres, keeping in mind generalisation to the whole data set. SVM were found to perform similar to RBF networks, but required much more information and subsequently more resources for their computation. It could be concluded that the classes are not particularly better described by their boundaries, especially when the whole dive is considered.

It is clear that the 50 first clicks of a click train do not accurately represent the variability in the data, which makes it difficult to generalise to the whole data set. From the current data it is not clear how much variability comes from changes in background noise and how much can be ascribed to a change in orientation of the whale. However, it is clear that the RBF network does have the capacity to classify the data once more patterns become available for training. A future algorithm might use a kind of adapting network, where the centres are moved in the direction of newly classified patterns. This could be done with a decaying learning parameter to limit their movements. A problem could be that current classification showed possible burst errors in Figure 8.3. While occasional random errors would not affect an adapting network too much, burst errors could destroy all reliability in the outputs. More data will be required to understand the evolution of the features, to assess what kind of variability can be expected and to see if this can be accounted for in the parameters of the network (perhaps adding a few kernels, or using larger standard deviations), or if on-line adaptation of the parameters will be required.

Chapter 9

Initialisation of the Classifier

9.1 Introduction

The classification algorithms used in the previous chapters relied on a training set with separated classes to initialise the algorithms' parameters. The training set only relies on a small portion of the data which could be manually analysed, but obviously it is preferred (or even required when the classification is performed from an autonomous monitoring platform) to automate the whole process. The automatisation would need to be able to both estimate the number of animals in a recording and to separate the first few clicks. Using the knowledge from Chapters 7 and 8 that showed and used the clustering tendency of the features, it would make sense to try a clustering algorithm for initial separation and preferably one that does not need to know the number of classes beforehand. One such algorithm is the use of a variational mixture of Gaussians. It is similar to training a Gaussian mixture model, optimising its log-likelihood through expectation maximisation, with the addition that distributions are placed on the mixture parameters. This automatically gives a way to evaluate the number of mixtures that are required by pruning those that have insignificant mixing coefficients and avoids problems with the likelihood optimisation when mixtures are placed on top of a data point.

9.2 Gaussian Mixture Model

This section aims to give a broad overview of the method that could be used to initially separate the sperm whale clicks. Details can be found in [16]. Since Gaussian kernels worked fairly well in the previous chapters, it was reasonable to continue with the radial basis function model, but in the form of a Gaussian mixture model (GMM). The goal has now changed from purely classifying the animals to estimating their number. From the RBF analysis with c-means clustering in Section 8.3.1 it was already found that when all data is clustered together it can be expected that between 2 or 3 clusters per animal are necessary and this will allow to estimate the number of animals. The clustering can be allowed to take more time as it is only done once at the start of a recording. The difference between the GMM and RBF will be that the GMM will use probability densities and can give a likelihood estimation for the number of clusters for model selection. Often, when clusters are added to a clustering algorithm like k-means, the likelihood of the data under that model also increases. Various criteria exist that take into account the number of free parameters available and use that as a penalty to find the best model. For the GMM approach here, the optimisation routine will lead to small weights for kernels that have little relevance, allowing them to be pruned.

The starting point is the Gaussian mixture :

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k), \tag{9.1}$$

$$0 \leq w_k \leq 1, \quad \sum_{k=1}^{K} w_k = 1. \tag{9.2}$$

This equation describes the linear sum of normal distributions, as is done with a single node for the RBF network, except that here there is no longer an assumption that the covariance of all kernels is equal. The key is to maximise the likelihood of the data with respect to the parameters of the Gaussian mixtures :

$$p(\mathbf{X}|\mathbf{w}, \mu, \mathbf{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k) \tag{9.3}$$

where $\mathbf{X}$ is the data matrix with rows $\mathbf{x_n}$ and N the number of samples. Maximisation of the log likelihood of (9.3) will be difficult due to the sum that appears inside the logarithm. One way to simplify the optimisation is by the use of a latent variable $\mathbf{z_n}$, $z_{nk} \in \{0, 1\}$ and $\sum_k z_{nk} = 1$, that assigns each data point $\mathbf{x}_n$ to a mixture. The distribution on $\mathbf{z_n}$ can be written as

$$p(\mathbf{z}_n|\mathbf{w}) = \prod_{k=1}^{K} w_k^{z_{nk}} \tag{9.4}$$

and the probability of a point $\mathbf{x}_n$ given a $\mathbf{z}_n$

$$p(\mathbf{x}_n|\mathbf{z}_n, \mu, \mathbf{\Sigma}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)^{z_{nk}} \tag{9.5}$$

The likelihood function of the complete data set $\{\mathbf{X}, \mathbf{Z}\}$ then becomes

$$p(\mathbf{X}, \mathbf{Z}|\mathbf{w}, \mu, \mathbf{\Sigma}) = \prod_{n=1}^{N} \prod_{k=1}^{K} w_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)^{z_{nk}} \tag{9.6}$$

or the log-likelihood

$$\ln p(\mathbf{X}, \mathbf{Z}|\mathbf{w}, \mu, \mathbf{\Sigma}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\ln w_k + \ln \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k) \right). \tag{9.7}$$

This equation basically evaluates the likelihood for each data point $\mathbf{x}_n$ only for its own mixture (when $z_{nk} = 1$). Maximisation of the log-likelihood has been simplified by eliminating the sum in (9.3), but of course $\mathbf{Z}$ is not known and has to be evaluated itself. This is usually done by computing the expectation of the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \mathbf{w}, \mu, \mathbf{\Sigma})$. Since the parameters $\{\mathbf{w}, \mu, \mathbf{\Sigma}\}$ appear both in the posterior distribution for z_{nk} and under the logarithms on the right side of equation (9.7) direct optimisation can be difficult. Therefore it is usually done in two steps with the expectation-maximisation algorithm, first the expectation of the posterior distribution is updated using old parameters (or an initial guess), then (9.7) is maximised keeping $\mathtt{E}(z_{nk})$ fixed.

9.2.1 Bayesian Approach

One problem with the log-likelihood function (9.7) is the possibility of a kernel being placed exactly on top of a data point. On the right hand side there will appear a term $|\Sigma|^{-1}$ which can grow arbitrarily large for shrinking variance. As long as all data points are covered by at least one other kernel with a broad enough variance so that they will not contribute a zero value to the log-likelihood, then this cannot be immediately prevented. One way to handle this problem is to detect when kernels are on top of a data point, and in that case to relocate it randomly while resetting its variance. As the right value for K is not known and will be overestimated, this problem will certainly occur for many of the kernels and will not be practical.

Another way to deal with this is to define a prior distribution over the variance. As both the mean and variance of the Gaussian mixtures are unknown, a Gaussian-Wishart prior distribution can be defined over these unknown parameters :

$$p(\mu, \mathbf{\Sigma}) = p(\mu|\mathbf{\Sigma})p(\mathbf{\Sigma}) = \prod_{k=1}^{K} \mathcal{N}(\mu_k|\mu_0, \beta_0\mathbf{\Sigma}_k)\mathcal{W}(\mathbf{\Sigma_k}|\mathbf{V}_0, \nu_0). \tag{9.8}$$

It is noted that these are conjugate prior distributions, meaning that the posterior distributions of the parameters taking into account X have the same form, which can make computations much easier.

The real posterior distribution on the latent variables and mixture parameters is unknown, but will be inferred through a distribution $q(\mathbf{Z}, \mu, \mathbf{\Sigma})$, assuming that the latent variables and parameters are independent this can be split into $q(\mathbf{Z})q(\mu, \mathbf{\Sigma})$. These two distributions can be updated in sequence, using result (10.9) from [16]. For $q(\mathbf{Z})$ it follows that for the optimal q^* :

$$\begin{aligned} \ln q^*(\mathbf{Z}) &= \mathbb{E}_{\mu,\mathbf{\Sigma}}[\ln p(\mathbf{X}, \mathbf{Z}, \mu, \mathbf{\Sigma}|\mathbf{w})] \\ &= \mathbb{E}_{\mu,\mathbf{\Sigma}}[\ln (p(\mathbf{X}|\mathbf{Z}, \mu, \mathbf{\Sigma}, \mathbf{w})p(\mathbf{Z}|\mathbf{w})p(\mu, \mathbf{\Sigma})p(\mathbf{\Sigma}))] \\ &= \mathbb{E}_{\mu,\mathbf{\Sigma}}[\ln p(\mathbf{X}|\mathbf{Z}, \mu, \mathbf{\Sigma}, \mathbf{w})] + \ln p(\mathbf{Z}|\mathbf{w}) + \text{const} \end{aligned} \tag{9.9}$$

where the constant includes all terms that do not depends on Z. Using equations (9.4) and (9.5) this leads to

$$\begin{aligned} \ln q^*(\mathbf{Z}) &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\{\ln w_k + \frac{1}{2}\mathbb{E}[\ln|\mathbf{\Sigma}_k|] - \frac{D}{2}\ln(2\pi) \\ &\quad -\frac{1}{2}\mathbb{E}_{\mu_k,\mathbf{\Sigma}_k}[(\mathbf{x}_n - \mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n - \mu_k)]\} + \text{const} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln\rho_{nk} + \text{const}. \end{aligned} \tag{9.10}$$

Defining normalised coefficients $r_{nk} = \rho_{nk}/\sum_k \rho_{nk}$ this gives the optimal posterior distribution over Z as

$$q^*(\mathbf{Z}) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}} \tag{9.11}$$

with again a multinomial distribution as the prior in (9.4) and $\mathbb{E}[z_{nk}] = r_{nk}$. Similarly, the optimal posterior distribution $q(\mu, \mathbf{\Sigma}) = \prod_k q(\mu_k, \mathbf{\Sigma}_k) = \prod_k q(\mu_k|\mathbf{\Sigma}_k)q(\mathbf{\Sigma}_k)$ has the same form as its prior :

$$q^*(\mu_k, \mathbf{\Sigma}_k) = \mathcal{N}(\mu_k|\mu_k, \beta_k\mathbf{\Sigma}_k)\mathcal{W}(\mathbf{\Sigma_k}|\mathbf{V}_k, \nu_k). \tag{9.12}$$

The E-step of the expectation maximisation algorithm consists of evaluating the expectations in (9.10), giving

$$\mathbb{E}[\ln|\mathbf{\Sigma}_k|] = \sum_{d=1}^{D}\psi\left(\frac{1}{2}\{\nu_k + 1 - d\}\right) + D\ln 2 + \ln|\mathbf{V}_k| \tag{9.13}$$

$$\mathbb{E}_{\mu_k,\mathbf{\Sigma}_k}[(\mathbf{x}_n - \mu_k)^T\Sigma_k^{-1}(\mathbf{x}_n - \mu_k)] = D\beta_k + \nu_k(\mathbf{x}_n - \mathbf{m})^T\mathbf{V}_k(\mathbf{x}_n - \mathbf{m}) \tag{9.14}$$

where ψ is the digamma function and D the dimension of the input space. Evaluation of the optimal posterior distributions follows directly for $q^*(\mathbf{Z})$ through computation of the coefficients r_{nk}. For $q^*(\mu, \mathbf{\Sigma})$ the following update equations can be deduced [89, 16] :

$$\begin{aligned}
\beta_k^{-1} &= \beta_0^{-1} + N_k \\
\mathbf{m}_k &= \beta_k(\beta_0^{-1}\mathbf{m}_0 + N_k\overline{\mathbf{x}}_k) \\
\mathbf{V}_k^{-1} &= \mathbf{V}_0^{-1} + N_k\mathbf{S}_k + \frac{N_k}{1+\beta_0 N_k}(\overline{\mathbf{x}}_k - \mathbf{m}_0)(\overline{\mathbf{x}}_k - \mathbf{m}_0)^T \\
\nu_k &= \nu_0 + N_k \\
\text{with} & \\
N_k &= \sum_{n=1}^{N} r_{nk} \\
\overline{\mathbf{x}}_k &= N_k^{-1}\sum_{n=1}^{N} r_{nk}\mathbf{x}_n \\
\mathbf{S}_k &= N_k^{-1}\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \overline{\mathbf{x}}_k)(\mathbf{x}_n - \overline{\mathbf{x}}_k)^T. \qquad (9.15)
\end{aligned}$$

(9.16)

9.3 Algorithm Performance

Using the prior distribution on the Gaussian mixture parameters requires initial initialisation of a number of hyper-parameters. The current data set was not large enough to see if there were some global values that can be expected to always perform well. For the purpose of evaluating a Gaussian mixture model in order to estimate the number of vocal animals, and to create an initial training set, the parameters were not tested over a large interval. Clustering was performed on the first 50 clicks of a click train, using the 15 most discriminating features that were selected by the local discriminating basis. To initialise the parameters, $\mathbf{m}_0$ was set to $\mathbf{0}$ while m_k was set by taking random data from the selected clicks. Furthermore, $\beta_k = \beta_0 = 1$, $\mathbf{V}_k = \mathbf{V}_0 = \mathbf{I}$ and $\nu_k = \nu_0 = D + N$. The number of clusters in the mixture K was set to 20. Clusters were considered significant after optimisation had converged if N_k was at least 20. This was set considering that one minute of data should contain roughly 50 clicks of a sperm whale, and knowing from RBF training that each whale can be described with 2 to 3 clusters, it should be expected that a mixture describing a single animal should cover at least 20 data samples.

The clustering results are shown in Figure 9.1 for four data sets containing 4 to 7 animals. Each clustering was repeated 100 times to account for the randomness in the initial assignment of $\mathbf{m}_k$ and the local optima.

It can be seen that there is a tendency to underestimate the real number of animals in the cluster. All four times the bin that estimated one animal less than the real number scored high, and even higher half the times. This could be adjusted by fine-tuning different parameters, most obviously the counter based on N_k. Without more training data it will be difficult to find a value that can be expected to generalise well. The fact that each animal appears to have a single strong cluster associated to it (otherwise there would have been a larger over estimation of the number of animals) will make it easier to construct an initial training set for the RBF classifier. When multiple clusters are covering an animal, the distance between the cluster centres could be evaluated to decide whether or not two clusters describe data from the same animal, but it appears that this is not necessary when the parameters are further fine-tuned. Underestimation seems more likely to occur than overestimation.

9.4 Conclusion

Creating a training set for the radial basis classifier automatically from a recording requires an unsupervised algorithm that can estimate the number of animals and make an initial separation. It seems that a good candidate algorithm is a variational mixture of Gaussians. This algorithm avoids problems with maximum likelihood estimators when a kernel is placed on top of a data point and is capable of reducing the number of kernels when they have little relevance. On the available data it gave a reasonably accurate estimate of the number of animals presented to it, with a preference to underestimate the true number. Especially promising was that it estimated the number of animals using a single significant mixture (explaining more than 20 clicks) per animal. If this behaviour remains consistent on other data sets then it would automatically solve the second part of separating the data for an initial data set. Another option would have been to investigate the distance between mixture centres, but it appears unnecessary.

Each estimate did require considerable computation time compared to classification of a pattern under RBF, as the clustering process had to be repeated many times to obtain a reliable solution due to the randomness in the initialisation and local optima. This will generally not be a problem, an application can buffer incoming data while the GMM is working on creating a training set. Once the RBF classifier has been trained it can easily catch up with the buffer. In order to occasionally update RBF parameters to allow it to adjust to trends in the features or changing noise patterns, the GMM can be run in the background on spare CPU cycles.

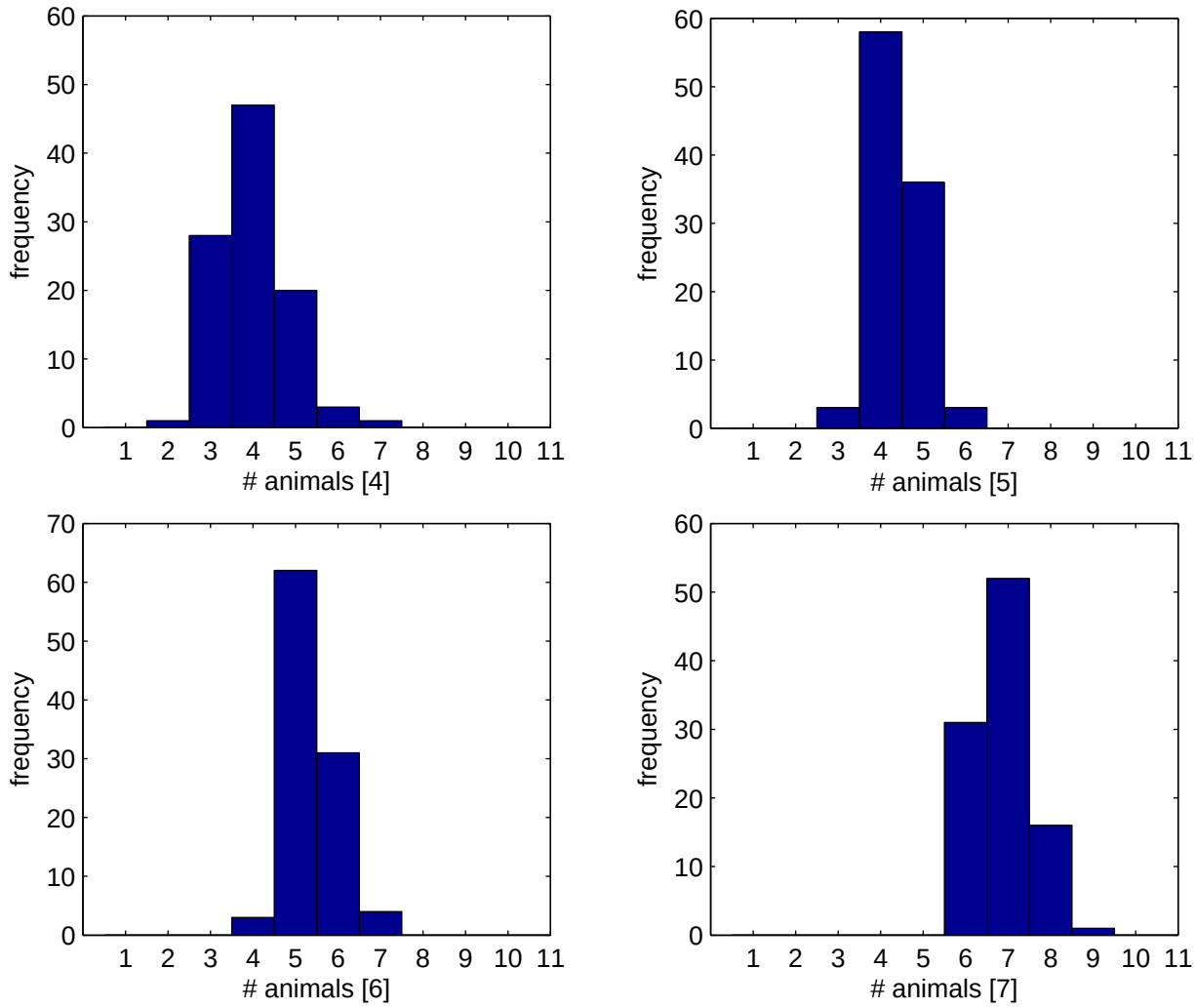

Figure 9.1: Results of unsupervised clustering with a variational mixture of Gaussians to estimate the number of animals. The real number of animals is given between brackets on the x-axis. The algorithm seems to have a preference for underestimation of the true number.

Chapter 10

Discussion

10.1 Discussion of the results

The premise of the thesis was to design an algorithm that could be capable of distinguishing between a number of animals in real time and minimal training data. Here the assumption was (from personal experience) that in a typical recording up to five animals might be heard, and more than five would be rare. For that reason a data set of seven animals was used. Originally it was hoped that a typical example click could be found, or perhaps a small set of clicks, that would form an unique identifier for an animal, using cross correlation on the time domain signals for their separation. After it was found that this only worked on small and very limited sequences, it became clear that the search for this unique identifier had to be expanded to the time-frequency domain. Combining research from Goold on dominant frequencies and from Kamminga that a narrow band selection of dolphin sonar can be described accurately by a Gabor function, a Gabor model was fit on dominant sperm whale frequencies. It has been shown that this approach was not successful due to very specific characteristic of the clicks. Both cross correlation on the time domain signal and modelling of a single frequency did not only fail to provide a unique finger print of an animal, but also generally failed at separating animals within one recording.

As there was, and still is, some interest in the function of codas as an identifier of a sperm whale group as a unique social unit, some attention was given to identifying coda rhythms, although this did not constitute a bio-metric in the sense of e.g. a fingerprint. It was very difficult to find reliable reports on the number and types of codas that were found for different social units. Most coda labelling was done subjectively and lack of information concerning the variance of a cluster made it difficult to assess if the label truly described a specific unique class, or included many patterns that could also fall into other neighbouring classes. For this reason, a more objective labelling protocol was defined and published. If adopted by other researchers, some conclusions concerning the uniqueness of a pattern could be made in the future, but at the moment of writing the assumption that a single coda rhythm might identify a sperm whale social unit seems optimistic. Certainly, more data needs to be analysed. For the purpose of an acoustic bio-metric, codas were no longer considered, especially since the main interest for the application is to separate the animals during their dive when they do not produce codas.

In the mean time, more evidence became available that the sperm whale sonar signal was highly directional, and the recorded clicks were highly dependent on the relative angle between the hydrophone and the animal. After this discovery it became clear why the correlation and Gabor function algorithms failed and that they would likely remain unsuccessful when the recording angle is unknown. If, for example, on axis recordings were available from multiple animals, then separation with a Gabor function would again become

an interesting option. But as these recordings are extremely rare, especially in situations where recordings are made from a boat, it was important to find a more robust method. This directional influence also meant that the creation of a library of unique acoustic identifiers for sperm whales, as is done for visual identification with dorsal fins, would be unlikely. At the same time, separation of animals during one recording, which poses much lighter requirements on the uniqueness of the identifiers as it only requires distinguishing between roughly five animals at one time, could still be possible.

As mentioned before, once the decision was made to search for characteristics from coefficients in the combined time-frequency domain there were many options for expressing the signal. A mathematical model of a sperm whale click was lacking (and is still missing today), and a priori there was no reason to prefer one method over another. It was considered important that the time-frequency representation would give ample choice between features, while at the same time they needed to be easily producible. It was found that the discrete wavelet packet transform satisfied these requirements. The discrete transform itself is a simple filter operation leading to a straightforward implementation in software or directly in hardware, while the wavelet packet table gives sufficient flexibility in the choice of coefficients. This feature generation approach proved successful in finding characteristic information in the clicks when a small number of animals were considered.

It was interesting to see that the best coefficients were generally found at the lowest frequencies, where the Gabor model still failed, but perhaps not surprising. This was probably due to a combination of the qualities of the signal itself and the signal treatment. On the one hand, higher frequencies in general are more influenced by directivity and propagation affects and could be considered more variable than lower frequencies. On the other hand, the preprocessing of the data synchronised the clicks on their shape in the time domain, focussing on the more stable lower frequencies. If instead it was tried to focus on specific higher frequency peaks then perhaps these could have been considered as features. But as was indicated in e.g. Figure 4.2, high frequency peaks could move around in the signal and the cross correlation function had many peaks where it was not clear which ones should be aligned. Another source of 'algorithmic noise' could have come from the aliasing in the wavelet packet table. If the high frequency components with different phase delays in a sequence of a few subsequent clicks are not filtered out, they will continue to influence the outputs of the low-pass frequency bands in the wavelet filter bank and can appear as noise. Therefore, after the identification of discriminating features, data was separately low-pass filtered during preprocessing to improve the performance. Of course, when in the end only such a small part of the spectrum is used, the use of a wavelet packet transform, even when it is only used for training a classifier, may seem exaggerated and unnecessary. Nevertheless, due to its flexibility and speed it is suggested to maintain it as part of the algorithm. While currently it seems that low frequency coefficients perform good, in the future it may be found that the data need to be preprocessed favouring high frequencies and then there is no reason to suspect that following the same approach with the creation of a discriminating basis through a wavelet packet table would not give similar results.

Using a simple linear separator directly on the discriminating coefficients was shown to still perform somewhat poor. Again, there were many different classification algorithms that could be looked at to improve this performance. Inspection of the feature space, however, showed a clustering tendency that could be exploited by placing distributions on the data in order to model those clusters. Considering speed of execution, the radial basis function network was a prime candidate. Training of the proposed algorithm does require a clustering process, which may be time intensive, but once the classifier has been trained evaluation of new patterns can be done very fast. In comparison with a support vector machine classifier, it was shown that the feature space was better described by modelling the areas of high density than by modelling the (projected) boundaries between the classes. Even though the RBF and SVM classifiers could perform similarly, the amount of information that was used for the classification was considerably less for RBF than for SVM. This could be caused by the limited training sets that were taken from the start of click sequences. As was shown in Figure 7.4, there might be trends in the data that are longer than the 50 clicks that were used

for training. This might be enough to define the sample mean of the feature, allowing a centre to be placed on top of it, but might poorly define the boundary of the class. Allowing misclassifications of the training data through regularisation parameters in SVM matched RBF performance but led to many support vectors.

An important point that remains is the initialisation of the classifier. The training algorithm for RBF requires supervision of the patterns. An unsupervised clustering algorithm was investigated based on the presented results to produce the training data. As training is only done once at the beginning of a recording, and then perhaps repeated occasionally during the processing of the data, it can be allowed to use more resources and to be slower than the actual classifier. Clustering for the RBF algorithm found that generally two normally distributed clusters could be sufficient to describe one animal. Thus a first estimation on the number of animals that are present in a recording could be made based on the number of significant clusters. Considering that in general clustering algorithms describe the data better when more clusters are used, a criterion was needed to find the optimal number of clusters. A Bayesian approach to train a mixture of Gaussians allowed to approximate the number of animals present in the recordings, with a preference for slight underestimation. This could probably be solved by adjusting the decision threshold of the algorithm. Especially interesting was that only one significant cluster was found for each animal while testing with different data combinations. This could indicate that the algorithm will also be able to perform the initial separation for the creation of a training set. While it is too early to conclude that this approach will always be able to initialise a subsequent classifier, it did show interesting properties worth of future research.

10.2 Continuation of the research

The capability of discriminating between individual sperm whales has been shown, but more work needs to be done to take into account all available information and robust initialisation of the classifier. Another important point will be the detection of a new animal unknown to the network. Ideally, there would be low values on all network outputs, as an RBF network gives a localised output (i.e. it does not generalise well at increasing distances from the hidden layer centres). But this cannot be guaranteed and needs to be further investigated. The purpose was to find an acoustic bio-metric, and as such, and after discarding codas, the classifiers did not take into account information on the inter-click intervals. Sperm whales tend to change their click production rate in a fairly smooth way. Regular clicks can be produced around 1 second intervals, while creaks can have 50 clicks per second, but changing from one into the other always follows a reasonably smooth curve. Clearly, including this kind of inter-click information can improve performance on classification, and a collaboration with the university of Pompeu Fabra is in progress to exploit this.

Besides acoustic or rhythmic information, when multiple hydrophones are available then information on the direction of arrival can be included in the algorithm. The platforms under consideration here often have a small hydrophone array installed, however these arrays can give a large error in the positional estimation. Thus this information by itself will not always be enough to separate animals, but obviously it will be very helpful for both the creation of a training set and the final classification.

Once more data becomes available, there can certainly be reason to look at other improvements of the data processing and classification. From synchronising the clicks on a different frequency band to changing the time-frequency representation to find better features. Once a model is found that accurately describes a click as a function of an animal's morphology, much better features may be found. But based on the presented research, the proposed method is considered suitable for implementation at the monitoring platforms that are currently being developed, especially those that are not powered by a cable and need to function autonomously for a long period of time with minimal power consumption. Cabled systems with access to virtually unlimited computational power might directly make use of for example the Bayesian clustering algorithm for classification of all patterns.

Chapter 11

Conclusion

Concerning the objectives of the thesis, the following conclusions can be drawn,

1. **Investigate the use of codas for sperm whale group identification**

   A coda labelling protocol has been designed that eliminates subjectivity and will allow comparison of codas between different studies. While at this moment it is not obvious if sperm whale groups can be identified by their coda rhythms, the protocol may help in this process.

2. **Investigate the Gabor Function as a model for dominant clicks in the sperm whale signal**

   It was found that the Gabor function is not a reliable model for the lowest dominant frequency in the sperm whale click. It is suspected that this is especially due to the influence of the orientation of an animal, leading to different time delays of pulses within the click.

3. **Find a method to extract discriminating features from sperm whale clicks**

   It was shown that a local discriminant basis can effectively discover discriminating features in sperm whale click signals, although here too the animal's orientation affected the algorithm.

4. **Find a classifier that allows distinguishing between sperm whales based on acoustic cues in real-time**

   A radial basis function network has shown good separability between 7 animals and reasonable generalisation towards data of a full dive. The network can be implemented in real-time without any problems due to its simple structure.

5. **Propose an unsupervised algorithm that can reliably create a training data set**

   The use of a Gaussian Mixture Model showed capability to estimate the number of animals in the group and to create an initial training set for the classifier.

The presented classification algorithm is considered to be suitable for implementation at autonomous acoustic monitoring platforms as a basis to perform real-time separation of foraging sperm whales in the area.

Appendix A

Description of Data

A.1 Click train data

The sperm whale data were collected from an inflatable boat during four field seasons spanning four to ten weeks each (from 1997 to 1999) at Kaikoura, New Zealand [47]. Recordings were made of solitary diving male sperm whales using an omni-directional hydrophone (Sonatech 8178; frequency response 100 Hz to 30 kHz $\pm$ 5 dB) lowered to a depth of 20 m. This hydrophone was first connected to a fixed gain amplifier (flat response from 0 to 45 kHz) and then to one channel of a Sony TCD-D10 PROII Digital Audio Tape recorder (frequency response 20 Hz to 22 kHz $\pm$ 1 dB with an anti-alias filter at 22 kHz). The recordings were digitised at 48 kHz and 16 bits. The use of data from solitary diving whales guarantees that no data from different animals were mixed, thus helping to obtain optimal results for the classification algorithms.

Data from seven different animals were available for this study, comprised of six single click trains and a complete dive. The typical duration of the recorded click trains was around 2.5 minutes. The complete dive covered around 30 minutes and was a sequence of such click trains. This dive was considered to be especially interesting as it allowed to see the performance of the algorithms, and validity of features, for the duration of an entire dive. Therefore, the dive was split up in two unequal parts. One click train early in the dive was separated and joined together with the other six available click trains used for training. The remainder of the dive was put in its own set denoted as 7* and was only used to test the classifiers, it was never used for training or parameter selection. This approach simulated the situation where a classifier would have to be trained with data at the start of a recording and allowed to assess its capacity to generalise to patterns much later in the dive sequence, that may have undergone changes as in Figure 4.5.

Initially, classification using the Gabor function or the simple linear classifier directly based on wavelet packet coefficients was tested on just 5 sets. These sets were not selected using any specific criterion, except that the long dive was not considered. For the more successful classifiers all click trains (7) were used for training and parameter selection, while the remaining set of the dive (7*) was exclusively used as a test validation set.

A.2 Coda data

The coda study done in Chapter 5 was based on data recorded in the period of 1993 to 1996 between the main islands of Gran Canaria, Tenerife and Fuerteventura [6]. A two-hydrophone array was used for the

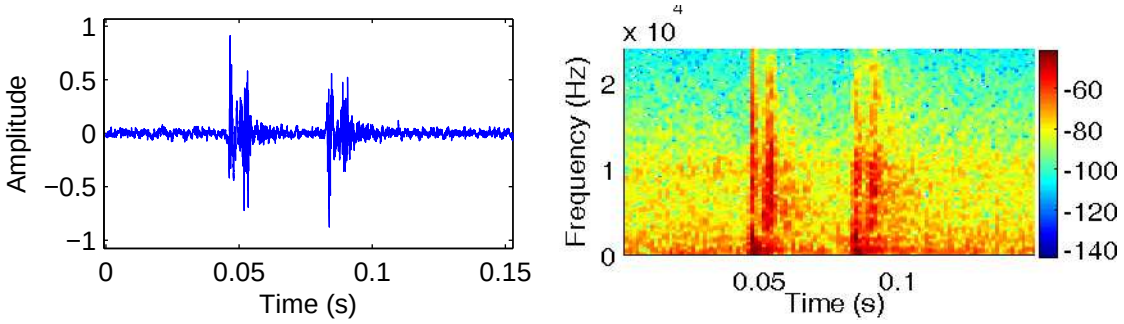

Figure A.1: The left image shows a typical example click, in an a-typical quiet environment, followed by a surface echo (verified by phase inversion). The right image shows its frequency spectrum. The second pulse that is visible within the click and its surface echo is likely an echo inside the head.

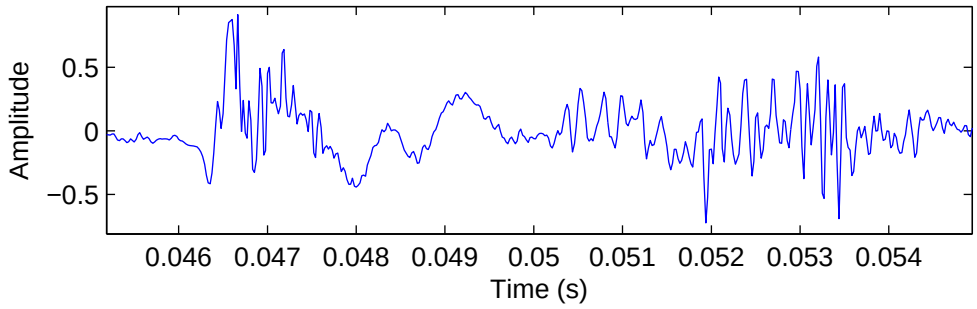

Figure A.2: The click at 0.05 s in Figure A.1 amplified to show its structure.

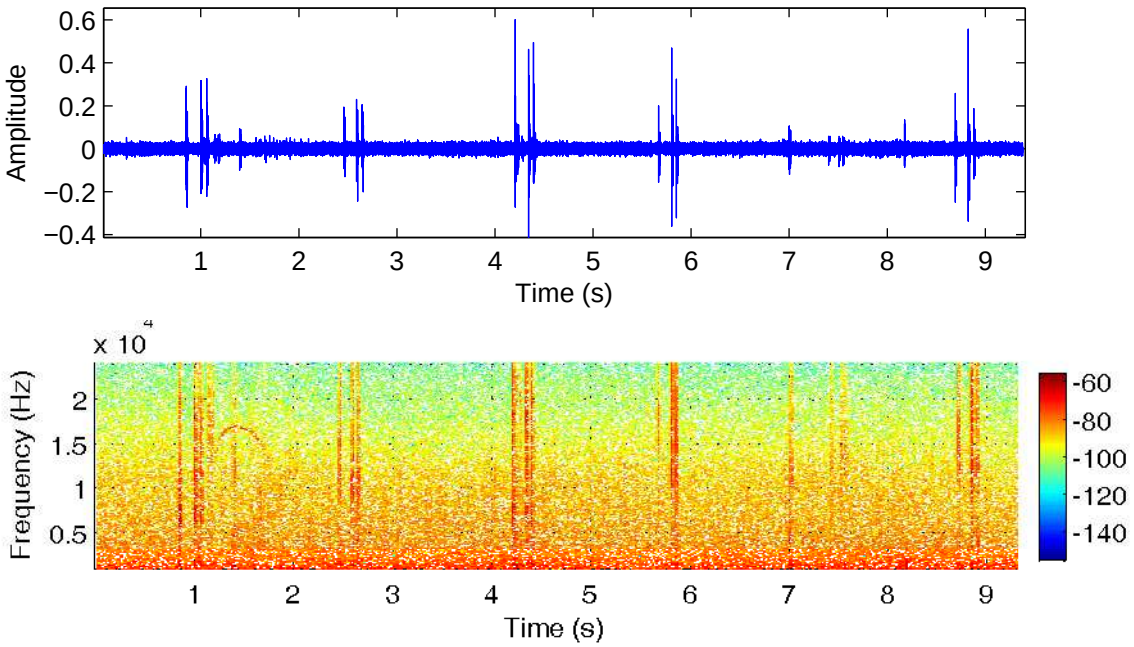

Figure A.3: Example of a coda repeated six times in 10 seconds. In the top figure the coda can be seen near seconds 1, 2.5, 4.2, 5.7, 7.5, 8.7. The bottom shows its corresponding spectrogram, note that around second 7 there is an impulse, but it is not part of the repeating coda, while the weak impulses around second 7.5 are. The weakness of this coda may be caused by source directivity.

recording, consisting of 2 Benthos AQ-4 elements (frequency response: 1 Hz - 15 kHz;32 dB ± 0.2 dB gain). The data was stored on a Sony PCDII Pro DAT recorder at 48 kHz sampling rate.

The codas were manually located in every recording and were kept if they had both a good signal to noise ratio and were clearly distinguishable from overlapping or successive codas. Additionally, they had to be repeated at least once to be considered for analysis. The start of the first pulse was taken as the start of the coda, defining the pulse intervals as the duration between the start of a pulse and the following pulse. The codas from all tapes were combined and grouped by the number of pulses and normalised by their duration.

Appendix B

Publications

B.1 Relevant publications in peer reviewed journals

van der Schaar, M., Delory, E. and André, M.
Classification of sperm whale clicks (*Physeter macrocephalus*) with Gaussian-kernel based networks.
Algorithms, vol 2, issue 3, p.1232-1247, 2009.

van der Schaar, M., Delory, E., Català, A. and André, M.
Neural network based sperm whale click classification.
Journal of the Marine Biological Association, 87:35–38, 2007.

van der Schaar, M., Delory, E., van der Weide, J., Kamminga, C., Goold, J., Jaquet, N. and André, M.
A comparison of model and non-model based time-frequency transforms for sperm whale click classification.
Journal of the Marine Biological Association, 87:27–34, 2007.

van der Schaar M. and André, M.
An Alternative Sperm Whale (*Physeter macrocephalus*) Coda Naming Protocol.
Aquatic Mammals, vol 32, issue 3, p.370-373, 2006.

B.2 Relevant talks

van der Schaar, M., Zaugg, S., Houégnigan, L., Castell, J.V., André, M.
Architecture for the Real-Time Monitoring of Noise Pollution and Marine Mammal Activity.
MARTECH 2009, Vilanova i la Geltrú, Spain, Nov 2009.

van der Schaar, M., Zaugg, S., Houégnigan, L., André, M.
Real-time processing and management of acoustic data streams in LIDO.
4th International Workshop on Detection, Classification and Localization of Marine Mammals Using Passive Acoustics, Pavia, Italy, Sep 2009.

van der Schaar, M., Delory, E., André, M.
Sperm whale (Physeter macrocephalus) acoustic identification
2th International Congress on Maritime Technological Innovations and Research, Barcelona, Nov 2007.

van der Schaar, M., Delory, E. and André, M.
Comparison of Gaussian based methods to classify sperm whale (*Physeter macrocephalus*) clicks.
Proceedings of the Sea Tech Week, Brest, France, Oct 2006.

van der Schaar, M. and André, M.
Demonstration of a Standard Method for Sperm Whale Coda Clustering.
Proceedings of the Sea Tech Week, Brest, France, Oct 2006.

van der Schaar, M., Delory, E., Van der Weide, J. and André, M.
Sperm Whale Click Identification.
7th French Workshop on Underwater Acoustics, Sea-Tech Week, Brest, France, Oct 2004.

van der Schaar, M., van der Weide, J., Kamminga, C., Goold, J., Jaquet, N., Delory, E. and André, M.
Dominant Frequencies in Sperm Whale Clicks.
17th Annual Conference of the European Cetacean Society, Las Palmas de Gran Canaria, 2003.

B.3 Relevant Posters

van der Schaar, M., Zaugg, S., Riccobene, G., Sánchez, T., Pubill, O., André, M.
System Architecture for Real-Time Monitoring of Noise Pollution
23th conference of the European Cetacean Society, Istanbul, Turkey, Mar 2009.

van der Schaar, M., Delory, E., André, M.
An adapting neural network for sperm whale separation.
22th conference of the European Cetacean Society, Egmond aan Zee, The Netherlands, Mar 2008.

van der Schaar, M., Delory, E. and André, M.
Support Vector Machine Methods Applied to the Classification of Sperm Whales
21th conference of the European Cetacean Society, San Sebastian, Spain, Apr 2007.

van der Schaar, M., Delory, E., Catal and A., André, M.
Neural network based sperm whale click classification.
20th conference of the European Cetacean Society, Gdynia, Poland, Apr 2006.

van der Schaar, M., Gallego, P. and André, M.
A standard method for sperm whale coda classification
19th Conference of the European Cetacean Society, La Rochelle, France, Apr 2005.

van der Weide, J., Kamminga, C., van der Schaar, M., Jaquet, N. and André, M.
An Automated Procedure to Extract Sperm Whale Click Sequences from Noisy Observations.
16th Annual Conference of the European Cetacean Society, Liège, Belgium, 2002.

B.4 Relevant Other

van der Schaar, M., Delory, E. and André, M.
Identifying Sperm Whales.
JMBA Global Marine Environment, issue 5, p.31, 2007.

Bibliography

[1] O. Adam. The use of the Hilbert-Huang transform to analyze transient signals emitted by sperm whales. *Applied Acoustics*, 67:1134–1143, 2006.

[2] J.A. Aguilar and ANTARES Collaboration. The data acquisition system for the antares neutrino telescope. *Nuclear Instruments and Methods in Physics Research A*, 570:107–116, 2007.

[3] R.A. Altes. Radar/sonar acceleration estimation with linear-period modulated waveforms. *IEEE Transactions on Aerospace and Electronic Systems*, 26(6):914–924, Nov 1990.

[4] T.W. Anderson and D.A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, Jun 1952.

[5] M. André. *Distribution and conservation of the sperm whales (Physeter macrocephalus) in the Canary Islands*. PhD thesis, University of Las Palmas de Gran Canaria, Spain, 1997.

[6] M. André, T. Johansson, E. Delory, and M. van der Schaar. Foraging on squid: the sperm whale mid-range sonar. *Journal of the Marine Biological Association*, 87:59–67, 2007.

[7] M. André and C. Kamminga. Rhythmic dimension in the echolocation click trains of sperm whales: a possible function of identification and communication. *Journal of the Marine Biological Association*, 80:163–169, 2000.

[8] C. Angulo, X. Parra, and A. Català. K-SVCR. A support vector machine for multi-class classification. *Neurocomputing*, 55:57–77, 2003.

[9] I.C. Ansmann, J.C. Goold, P.G.H. Evans, M. Simmonds, and S.G. Keith. Variation in the whistle characteristics of short-beaked common dolphins, *Delphinus delphis*, at two locations around the british isles. *Journal of the Marine Biological Association*, 87:19–26, 2007.

[10] E. Araki, K. Kawaguchi, S. Kaneko, and Y. Kaneda. Design of deep ocean submarine cable observation network for earthquakes and tsunamis. pages 1–4. Oceans 2008 - MTS/IEEE Kobe Techno-Ocean, April 2008.

[11] W. Au. A comparison of the sonar capabilities of bats and dolphins. In J.A. Thomas, C.F. Moss, and M. Vater, editors, *Echolocation in Bats and Dolphins*, pages xiii–xxvii. The University of Chicago Press, 2004.

[12] W.W.L. Au, A. Frankel, D.A. Helweg, and D.H. Cato. Against the humpback whale sonar hypothesis. *IEEE. J. Oceanic Eng.*, 26:295–300, 2001.

[13] R.H. Backus and W.E. Schevill. Physeter clicks. In K.S. Norris, editor, *Whales, Dolphines and Porpoises*, pages 510–528. Unif. Calif. Press, Berkely, 1966.

[14] R. Beitsma. Reverberaties in dolfijnsignalen. Master's thesis, Delft University of Technology, 1989.

[15] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford university press, 1995.

[16] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] S.B. Blackwell, J.W. Lawson, and M.T. Williams. Tolerance by ringed seals (phoca hispida) to impact pipedriving and construction sounds at an oil production island. *Journal of the Acoustical Society of America*, 115:2346–2357, 2004.

[18] M.J. Buckingham, J.R. Potter, and C.L. Epifanio. Seeing underwater with background noise. *Scientific American*, 274:40–44, 1996.

[19] R.-G. Busnel and A. Dziedzic. Observations sur le comportement et les émissions acoustiques du cachalot lors de la chasse. *Bocagiana, Museu Municipal do funchal*, 14:1–15, 1967.

[20] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7:536–541, 1930.

[21] M.C. Caldwell and D.K. Caldwell. Statistical evidence for individual signature whistles in pacific whitesided dolphins, *Lagenorhynchus obliquidens*. *Cetology*, 3:9, 1971.

[22] P. Chen, C. Lin, and B. Schölkopf. A tutorial on ν-support vector machines. http://www.csie.ntu.edu.tw/˜cjlin/papers/nusvmtutorial.pdf, 2003.

[23] T.F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

[24] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[25] R.B. D'Agostino and M.A. Stephens. *Goodness-Of-Fit Techniques (Statistics, a Series of Textbooks and Monographs)*. Marcel Dekker, 1986.

[26] Laboratori d'Aplicacions Bioacústiques. http://www.lab.upc.es, 2006.

[27] J.A. David. Likely sensitivity of bottlenose dolphins to pile-driving noise. *Water and Environment Journal*, 20:48–54, 2006.

[28] E. Delory, M. André, J.-L. Navarro Mesa, and M. van der Schaar. On the possibility of detecting surfacing sperm whales using others' foraging clicks. *Journal of the Marine Biological Association*, 87:47–58, 2007.

[29] R. Dewey. The venus ocean cabled observatory. *CMOS Bulletin*, 37(3):77–82, 2009.

[30] D.L. Donoho. De-noising by Soft-Thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[31] A.M. Dougherty. Acoustic Identification of Individual Sperm Whales (Physeter macrocephalus). Master's thesis, University of Washington, 1999.

[32] D.M. Drumheller. Uniform spectral amplitude windowing for hyperbolic frequency modulated waveforms. Technical Report NRL/FR – 7140-94-9713, Naval Research Laboratory (US), Jul 1994.

[33] O.D. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2001.

[34] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

[35] European parliament and the council of the European Union. Directive 2008/56/ec of the european parliament and of the council of 17 june 2008 establishing a framework for community action in the field of marine environmental policy (marine strategy framework directive). *Official Journal of the European Union*, L164:19–40, June 2008.

[36] L.N. Frazer and E. Mercado III. A sonar model for humpback whale song. *IEEE J. Oceanic Eng.*, 25:160–182, 2000.

[37] D. Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159:591–595, 1947.

[38] C. Gervaise, C. Ioana, A. Vernier, C. Veytizou, and Y. Stephan. Automatic detection and identification of marine mammal vocalises. In *Proceedings of the Sea Tech Week, Brest, France, 2006*, October 2006.

[39] J.C. Goold and S.E. Jones. Time and frequency domain characteristics of sperm whale clicks. *Journal of the Acoustical Society of America*, 98(3):1279–1291, September 1995.

[40] J.C.D. Gordon. *The behaviour and ecology of sperm whales off Sri Lanka*. PhD thesis, University of Cambridge, 1987.

[41] J.C.D. Gordon. A simple photographic technique for measuring the length of whales from boats at sea. *Rep. Int. Whal. Commn.*, 40:581–588, 1990.

[42] L.N. Guinee, K. Chu, and E.M. Dorsey. Changes over time in the songs of known individual humpback whales *(Megaptera novaeangliae)*. In R. Payne, editor, *Communication abd Behavior of Whales*, pages 59–80. Boulder, CO: Westview, 1983.

[43] G. Hamerly and C. Elkan. Learning the k in k-means. *Neural Information Processing Systems Foundation*, 2003.

[44] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, and H.H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. A*, 454:903–995, 1998.

[45] Q.Q. Huynh, L.N. Cooper, N. Intrator, and H. Shouval. Classification of Underwater Mammals Using Feature Extraction Based on Time-Frequency Analysis and BCM Theory. *IEEE Transactions on Signal Processing*, 46(5):1202–1207, May 1998.

[46] J.W. Ioup and G.E. Ioup. Self-organizing maps for sperm whale identification. In M. McKay and J. Nides, editors, *Proceedings: Twenty-Third Gulf of Mexico Information Transfer Meeting*, pages 121–129. U.S. Dept. of the Interior, Minerals Management Service, 2005.

[47] N. Jaquet, S. Dawson, and L. Douglas. Vocal behavior of male sperm whales: Why do they click ? *Journal of the Acoustical Society of America*, 109(5):2254–2259, May 2001.

[48] C. Kamminga and A.B. Cohen Stuart. Wave shape estimation of delphinid sonar signals, a parametric model approach. *Acoustics Letters*, 19(4):70–76, 1995.

[49] C. Kamminga and A.B. Cohen Stuart. Parametric modelling of polycyclic dolphin sonar wave shapes. *Acoustics Letters*, 19(12):237–244, 1996.

[50] V. Kandia and Y. Stylianou. Detection of sperm whale clicks based on the teager-kaiser energy operator. *Applied Acoustics*, 67:1144–1163, 2006.

[51] I. Katsavounidis, C.C.J. Kuo, and Z. Zhang. A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters*, 1(10):144–146, october 1994.

[52] D.W. Laist, A.R. Knowlton, J.G. Mead, A.S. Collet, and M. Podesta. Collisions between ships and whales. *Marine Mammal Sciences*, 17(1):35–75, 2001.

[53] C. Laplanche, O. Adam, M. Lopatka, and J.F. Motsch. Male sperm whale acoustic behavior observed from multipaths at a single hydrophone. *Journal of the Acoustical Society of America*, 118(4):2677–2687, October 2005.

[54] C. Levenson. Source level and bistatic target strength of the sperm whale (physeter catodon) measured from an oceanographic aircraft. *Journal of the Acoustical Society of America*, 55:1100–1103, 1974.

[55] M. Lopatka, O. Adam, C. Laplanche, J.F. Motsch, and J. Zarzycki. Sperm whale click analysis using a recursive time-variant lattice filter. *Applied Acoustics*, 67:1118–1133, 2006.

[56] P.T. Madsen, D.A. Carder, W.W.L. Au, B. Møhl, P.E. Nachtigall, and S.H. Ridgway. Sound production in neonate sperm whales. *Journal of the Acoustical Society of America*, 113:2988–2991, 2003.

[57] P.T. Madsen, M. Wahlberg, and B. Møhl. Male sperm whale (physeter macrocephalus) acoustics in a high-latitude habitat: implications for echolocation and communication. *Behavioral Ecology and Sociobiology*, 53:31–41, 2002.

[58] E. Massion and K. Raybould. Mars, the monterey accelerated research system. *Sea Technology*, 47:39–42, 2006.

[59] M. Mazzoil, S.D. McCulloch, R.H. Defran, and M.E. Murdoch. Use of digital photography and analysis of dorsal fins for photo-identification of bottlenose dolphins. *Aquatic Mammals*, 30(2):209–219, 2004.

[60] B. McCowan and D. Reiss. Quantitative comparison of whistle repertoires from captive adult bottlenose dolphins (delphinidae, tursiops truncatus): a re-evaluation of the signature whistle hypothesis. *Aquatic Mammals*, 100:194–209, 1995.

[61] E. Mercado III and L.N. Frazer. Humpback Whale Song or Humpback Whale Sonar ? A Reply to Au *et al. IEEE J. Oceanic Eng.*, 26:406–415, 2001.

[62] P. Miller, M. Johnson, and P. Tyack. Sperm whale behaviour indicates the use of echolocation click buzzes 'creaks' in prey capture. *Royal Soc. B*, 271:2239–2247, 2004.

[63] B. Møhl. Sound transmission in the nose of the sperm whale. *Journal of Comparative Physiology A*, 187:335–340, 2001.

[64] B. Møhl, P.T. Madsen, M. Wahlberg, W.W.L. Au, P.N. Nachtigal, and S.R. Ridgway. The monopulsed nature of sperm whale clicks. *ARLO*, 4(1):19–24, 2003.

[65] B. Møhl, M. Wahlberg, P.T. Madsen, A. Heerfordt, and A. Lund. The monopulsed nature of sperm whale clicks. *Journal of the Acoustical Society of America*, 114(2):1143–1154, August 2003.

[66] B. Møhl, M. Wahlberg, P.T. Madsen, P.T. Miller, and A. Surlykke. Sperm whale clicks: Directionality and source level revisited. *Journal of the Acoustical Society of America*, 107(1):638–648, 2000.

[67] J. Moody and C.J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 6(4):281–294, 1989.

[68] R.P. Morrissey, J. Ward, N. DiMarzio, S. Jarvis, and D.J. Moretti. Passive acoustic detection and localization of sperm whales *(Physeter macrocephalus)* in the tongue of the ocean. *Applied Acoustics*, 67:1091–1105, 2006.

[69] S.O. Murray, E. Mercado, and H.L. Roitblat. The neural network classification of false killer whale (Pseudorca crassidens) vocalizations. *Journal of the Acoustical Society of America*, 104(6):3626–3633, December 1998.

[70] M. Nishiwaki, S. Ohsumi, and Y. Maeda. Change of form in the sperm whale accompanied with growth. *Sci. Rep. Whales Res. Inst. Tokyo*, 17:1–13, 1963.

[71] K.S. Norris and G.W. Harvey. A theory for the function of the spermaceti organ of the sperm whale (physeter macrocephalus). In S.R. Galler, K. Schmidt-Hoenig, G.J. Jacobs, and R.E. Belleville, editors, *Animal Orientation and Navigation*, pages 397–419. NASA, Washington, D.C., 1972.

[72] S. Parsons and G. Jones. Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *The Journal of Experimental Biology*, 203:2641–2656, 2000.

[73] G. Pavan, T.J. Hayward, J.F. Borsani, M. Priano, M. Manghi, C. Fossati, and J. Gordon. Time patterns of sperm whale codas recorded in the mediterranean sea 1985-1996. *Journal of the Acoustical Society of America*, 107(6):3487–3495, 2000.

[74] K. Payne and R.S. Payne. Large scale changes over 19 years in songs of humpback whales in bermuda. *Z. Tierpsych*, 68:89–114, 1985.

[75] R. Person, L. Beranzoli, C. Berndt, J.J. Danobitia, M. Diepenbroecke, P. Favali, M. Gillooly, V. Lykousis, J.M. Miranda, J. Mienert, I.E. Priede, R.S. Santos, L. Thomsen, T. Van Weering, and C. Waldman. Esonet: An european sea observatory initiative. pages 1–6. OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean, April 2008.

[76] P. Phibbs, S. Mihaly, and R. Jones. System engineering at the edge of a cabled ocean observatory. pages 1–8. Oceans 2009 - Europe, May 2009.

[77] A.N. Popper. Sound emission and detection by delphinids. In L.M. Herman, editor, *Cetacean Behavior*, pages 1–52. Wiley, New York, 1980.

[78] A.N. Popper and M.C. Hastings. The effects of human-generated sound on fish. *Integrative Zoology*, 4:43–52, 2009.

[79] J.R. Potter, M.J. Buckingham, G.B. Deane, C.L. Epifanio, and N.M. Carbone. Acoustic daylight: preliminary results from an ambient noise imaging system. *Journal of the Acoustical Society of America*, 96(5):3235, November 1994.

[80] J.R. Potter and M. Chitre. Ambient noise imaging in warm shallow seas; second-order moment and model-based imaging algorithms. *Journal of the Acoustical Society of America*, 106(6):3201–3210, December 1999.

[81] L.E. Rendell and H. Whitehead. Comparing repertoires of sperm whale codas: A multiple methods approach. *Bioacoustics*, 14:61–81, 2003.

[82] L.E. Rendell and H. Whitehead. Spatial and temporal variation in sperm whale coda variations: stable usage and local dialects. *Animal Behaviour*, 70(1):191–198, 2005.

[83] J.L. Romeu. Anderson-darling: A goodness of fit test for small samples assumptions. Technical Report 3, RAC START, 2003.

[84] N. Saito and R.R. Coifman. Local discriminant bases. In *Mathematical Imaging: Wavelet Applications in Signal and Image Processing II*, volume 2303. Proc SPIE, 1994.

[85] T.M. Schulz, H. Whitehead, S. Gero, and L. Rendell. Overlapping and matching of codas in vocal interactions between sperm whales: insights into communication function. *Animal Behaviour*, 76:1977–1988, 2008.

[86] M.A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, Sep 1974.

[87] C.R. Sturtivant and S. Datta. Techniques to isolate dolphin whistles and other tonal sounds from background noise. *Acoust. Lett.*, 18(10):189–193, 1995.

[88] R. Suzuki and J.R. Buck. Extraction and tracking of bottlenose dolphin whistle contours. *J. Acoust. Soc. Am.*, 108:2635–2636, 2000.

[89] M. Svensén and C.M. Bishop. Pattern Recognition and Machine Learning, solutions to exercises web-edition. http://research.microsoft.com/en-us/um/people/cmbishop/PRML/prml-web-sol-2007-10-05.pdf, October 2007.

[90] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, third edition, 2006.

[91] A. Thode, D.K. Mellinger, S. Stienessen, A. Martinez, and K. Mullin. Depth-dependent acoustic features of diving sperm whales (*Physeter macrocephalus*) in the Gulf of Mexico. *Journal of the Acoustical Society of America*, 112(1):308–321, July 2002.

[92] F. Thomsen, D. Franck, and J.K.B. Ford. Characteristics of whistles from the acoustic repertoire of resident killer whales (orcinus orca) off vancouver island, british columbia. *J. Acoust. Soc. Am.*, 109(3):1240–1246, 2001.

[93] C. Tiemann, Thode A., J. Straley, K. Folkert, and V. O'Connell. Three-dimensional localization of sperm whales using a single hydrophone. *Journal of the Acoustical Society of America*, 120(4):2355–2365, October 2006.

[94] M. van der Schaar and M. André. An alternative sperm whale (*Physeter macrocephalus*) coda naming protocol. *Aquatic Mammals*, 32(3):370–373, 2006.

[95] M. van der Schaar, E. Delory, and M. André. Classification of sperm whale clicks *(Physeter macrocephalus)* with gaussian-kernel based networks. *Algorithms*, 2(3):1232–1247, 2009.

[96] M. van der Schaar, E. Delory, A. Català, and M. André. Neural network based sperm whale click classification. *Journal of the Marine Biological Association*, 87:35–38, 2007.

[97] M. van der Schaar, E. Delory, J. van der Weide, C. Kamminga, J.C. Goold, N. Jaquet, and M. André. A comparison of model and non-model based time-frequency transforms for sperm whale click classification. *Journal of the Marine Biological Association*, 87:27–34, 2007.

[98] W.A. Watkins. Acoustics and the behavior of sperm whales. In R.G. Busnel and J.F. Fish, editors, *Animal Sonar Systems*, pages 293–289. Plenum, New York, 1980.

[99] W.A. Watkins, K. Moore, and P. Tyack. Sperm whale acoustic behaviors in the southeast caribbean. *Cetology*, 49:1–15, 1985.

[100] L. Weilgart and H. Whitehead. Group-specific dialects and geographical variation in coda repertoire in south pacific sperm whales. *Behavioural Ecology and Sociobiology*, 40:277–285, 1997.

[101] L.S. Weilgart. *Vocalisations of the sperm whale, Physeter macrocephalus, off the Galapagos Islands as related to behavioural and circumstantial variables*. PhD thesis, Dalhousie University Halifax, Nova Scotia, 1990.

[102] C.R. Weir, A. Frantzis, P. Alexiadou, and J.C. Goold. The burst-pulse nature of 'squeal' sounds emitted by sperm whales (physeter macrocephalus). *Journal of the Marine Biological Association*, 87:39–46, 2007.

[103] H. Whitehead. Direct estimation of within-group heterogeneity in photo-identification of sperm whales. *Marine Mammal Science*, 17(4):718–728, 2006.

[104] H. Whitehead, M. Dillon, S. Dufault, L. Weilgart, and J. Wright. Non-geographically based population structure of south pacific sperm whales: dialects, fluke-markings and genetics. *Journal of Animal Ecology*, 67:253–262, 1998.

[105] H. Whitehead and L. Weilgart. Patterns of visually observable behaviour and vocalizations in groups of female sperm whales. *Behaviour*, 118:275–296, 1991.

[106] W. Whitney. Observations of sperm whale sounds from great depths. *Marine Physical Laboratory, Scripps Institute of Oceanography*, 1-9, 1968.

[107] W.X. Zimmer, P.T. Madsen, V. Teloni, Johnson M.P., and P.L. Tyack. Off-axis effects on the multipulse structure of sperm whale usual clicks with implications for sound production. *Journal of the Acoustical Society of America*, 118(5):3337–3345, 2005.