# Automatic Classification of Bird Behaviour on the basis of Accelerometer Data

Merijn de Bakker
0321907

Bachelor thesis
Credits: 9 EC

Bachelor Artificial Intelligence

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

*Supervisors*
Dr. M.W. van Someren
Informatics Institute

Prof. dr. ir. W. Bouten
Institute for Biodiversity and Ecosystem Dynamics

Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 24th, 2011

**Abstract**

Lately, a growing population of biologists and ecologists interested in animal behaviour has been logging the movements of animals. Such a logging of data using accelerometers and auxiliary sensors is particularly interesting if species are difficult to observe in their habitat. In order to extract and classify patterns from the collected sensor data, using machine learning techniques can be viable approach. However, preceding work has revealed that particular behaviours are often misclassified. Further, segmentation of time series into individual classes using existing segmentation techniques is problematic. The study this report is concerned with, applies machine learning to accelerometer time series of birds, with the general aim of studying useful approaches for feature extraction, classification and segmentation. The mentioned difficulties are addressed by means of different classification schemes on the one hand, and segmentation using output confidences on the other hand. Results indicate that for most classes, simple feature have good separating power, while difficult classes may gain from binary classification schemes. Further, a pilot of segmentation based on classifier output yield hopeful result. The results provide practical insight in useful approaches further research can be pointed towards.

**keywords:** time series classification, bird behaviour, accelerometer, segmentation, binary classification

**Acknowledgement**

Firstly, I would like to thank my supervisors Maarten van Someren and Willem Bouten for their helpful insights and comments. I would also like to thank Yaiza Dronkers for the cooperation and for annotating the data sets.

# Contents

# Chapter 1

# Introduction

In recent years, methods for studying animal behaviour has been extended with the logging and analysis of sensor data. The improved technology allows biologists to study behaviours of animals in circumstances that are difficult to set foot on. However, given the nature and the abundance of the resulting data, manual interpretation is often difficult and time consuming. Therefore, artificial intelligence methods can be a welcome approach for analysing animal behaviours automatically from sensor data. However, it is not yet fully clear to what extent machine learning techniques can be successfully used in this domain.

This report is an attempt to gain more insight in this question. The goal is to study whether using automatic processing of accelerometer data in biological research may be feasible. More precisely, on the hand of accelerometer signal of birds, the effect of annotation approach, different classification schemes, and the features representing the data is explored. Besides the application of artificial intelligence techniques, an important aspect of the project is also that the results are practically useful. Hence, this project is not purely pointed at artificial intelligence theory, but aims at formulating advice for future work on a specific domain.

This report is structured as follows. Preceding work and theoretical considerations of the domain of study are introduced in section 1.1 and further. From this introduction, the specific tasks in this project can be put forward. Chapter 2 describes the approach and implementation details. Results of the experiments are discussed in chapter 3. Finally, chapter 4 presents a recapitulation and implications for further research.

## 1.1 Background

Sensor data have been widely used in industrial and medical monitoring. Lately, a growing interest can be observed in studying humans gestures and gaits by means of sensors. Given the improved capabilities and decreased size of logging devices, also biologists have become interested in analysing sensor data. Logging of accelerometer data and auxilliary sensors is particularly interesting for species that are difficult to observe in their habitat, for example because

of the size or accessibility of the geographical area in which a species resides. (Sakamoto et al., 2009) Except for scientific knowledge acquisition, researchers may have interest in analysing sensor data for other reasons, including ecological conservation (e.g. Bom (2010), Wilson et al. (2008)) and cattle monitoring (e.g. Martiskainen et al. (2009) and Guo et al. (2009)).

Whatever the background of the interest in sensor data, manual processing of that data is often not a viable option. This may be due to the lack of knowledge about possible classes, or because of manually processing the data is too time consuming. In either case, machine learning techniques can be used in order to automate the process of pattern recognition. However, automatic recognition of patterns in the sensor data comes with several difficulties, which will be briefly turned to now.

### 1.1.1 Time Series Classification

Time series are characterised by a series of successive data points, and the time series concerned with in this thesis are three-dimensional acceleration time series. In many applications of analysis, one has to apply dimensionality reduction by replacing the series with a simpler or better descriptive structure. Finding a meaningful mapping from a selection of accelerometer frames to behavioural classes, and the finding of different behaviours within longer time series are of a particular interest.

#### Time series representation

One of the most often used techniques for pattern recognition in time series is Dynamic Time Warping, in which a warping cost is calculated on the basis of how well signals (possibly of different lengths) can be aligned in the time dimension. Instead of using the raw signal, there are scholars concentrating on symbolic representations. The idea behind such a symbolic representation is that language processing techniques can be used. Lin et al. (2003) for instance, convert time series to string of a given alphabet. This is done by dimensionality reduction via Piecewise Aggregate Approximation, and then discretise this easier series to a string using. This method, Symbolic Aggregate approXimation, has proved to be efficient for use in streaming data.

A different perspective is to analyse the characteristics of a signal and use these characteristics as a representation. Thus, then the objective of the time series analysis in recognition tasks is to find representations that are meaningful descriptors for more abstract categories, allowing comparison to other series. In order to achieve this objective, the general approach is to describe discrete time series statistically, or to fit models or functions to the signal. This is feasible since at particular scales the discrete data points are usually dependent. Possibilities for features extent from the time domain to the domain of wavelets. Selecting the right features are expected to improve classification results considerably. Later on, such features are discussed in more depth.

**Time series segmentation**

A related difficulty in time series recognition is to split the time series into homogeneous sections. Himberg et al. (2001) propose an unsupervised algorithm for segmentation. As in many instances, the solution presented makes use of cost-minimisation method, in which the cost function reflects to what extent a segment is internally heterogeneous. The greedy Global Iterative Replacement is proposed, iteratively splitting and concatenating segments while minimising the overall cost. This method is faster than purely top-down methods, but a drawback is that the algorithm needs to know the number of segments in advance or finds the optimal number of segments, which is not feasible in many situations. Another unsupervised approach is presented by Linaker and Niklasson (2000) and uses model vectors to represented a segment. For optimization, the algorithm applies vector quantization, namely Adaptive Resource Allocating Vector Quantization Network. A moving average smoothes data and compares it with the real data points. A node is added as a model vector if the moving average of the input and the model vectors has a mismatch greater than some $\delta$, and if the patterns is stable enough. An advantage of this approach is that it finds the segments dynamically and that it can be extended easily to higher dimensional data.

However, instead of using an unsupervised method for splitting the data sequence into segments, if a supervised model is present, it may be possible to use classification outputs to segment the signal. Each window then, is assigned a class that results from the trained model. A drawback of this approach is that the window may include a transition from one segment to another, since the windows usually cannot be arbitrarily small given the extracted features.

### 1.1.2 Automatic classification of Behaviour

As said above, automatic classification of accelerometer time series is not confined to the domain of animal behaviour. A brief literature survey reveals several research articles, both academic and commercial (mobile device companies), on human gesture recognition (see among others Bao and Intille (2004), Gyllensten (2010), Mantyla et al. (2000) and Pylvänäinen (2005)). The difficulty of recognising activities in real-life situations is acknowledged by Gyllensten (2010) In that study, a tri-axial accelerometer was used to answer the question how laboratory-collected data performs in classification contrasted with real-life situations. Besides of that, the difference in performance of three classifiers was evaluated. Indeed, the daily-life experiments showed a decrease (between 10% and 20%) in performance for all classifiers. This was partly because of the fact that similar behaviour can be performed in many different ways, which happens in real-life situations more often than under laboratory circumstances. In general, Support Vector Machines slightly outperformed the Decision Tree and Artificial Neural Network techniques, although these results varied according to the used feature sets.

Similar results have been found in other studies. Using Support Vector Machines, Martiskainen et al. (2009) classified several cow behaviours such as standing, sitting, feeding. Classification rates varied considerably among the

behaviours, with a classification accuracy of zero for the action of lying down to 80% for static lying and standing. Furthermore, it appeared the every class was often confused with at least one other class. In Sakamoto et al. (2009), European shags were equipped with a one-dimensional (surge) accelerometer logger. The collected data was processed using Wavelet-transforms. Windows of 1-second are applied to the wavelet spectra, and subsequently subjected to a k-means classifier. The resulting ethograms were evaluated by comparing the results with waterdepth profiles. Again, some classes, for example walking and running, were difficult to separate.

These findings correspond with findings of UvA-projects (Droeppelmann et al. (2011) and Hueting et al. (2011)) on classification of bird behaviours. Also there, particular high-level behaviours appeared to be difficult to classify on the basis of solely accelerometer data. Some general difficulties in the classification of behaviours can be identified. Firstly, particular high-level behaviours are often confused by the classifier. Furthermore, it could not clearly stated what features performed best on the whole range of behaviours. Lastly, the behaviour of birds, and in particular the oystercatcher, changes quickly and has different durations, which makes classification using fixed segments problematic.

In the context of acceleration time series representation, Preece et al. (2009) has performed a study to the relation between features and classification errors in the domain of accelerometer data classification on human activities. Preece et al. (2009) selected features from several scientific articles and trained models on several data sets. A distinction was made between time features, that is features extracted directly from the time series, frequency features, extracted using Fourier analysis, and wavelet features. The results in Preece et al. (2009) reveal that particularly the time and frequency features are informative, while the wavelet features yielded less high accuracies. A reason for this may be that the number of samples is too low for extracting the wavelet components conveniently. Further, these results cannot be simply extended to other situations since seemingly slightly different movements can result in very different accelerometer signals.

### 1.1.3 Problem definition

The theoretical background has provided some insight in the possibilities and difficulties of acceleration time series classification. It has become clear that the success of classifying such data depends on many factors, such as the domain of application, data representation and learning algorithm. Furthermore, finding segments corresponding to categories is problematic. In this thesis, these difficulties will explored with regard to the domain of acceleration data of birds, leading to the following problem definition:

*Given the domain of accelerometer data of bird behaviour, what are feasible machine learning approaches concerning data annotation, time series classification and segmentation?*

This question can be further explicated. The general purpose of the project is, given the domain of bird behaviour, to provide some fundamental insights in

the use of machine learning for studying accelerometer data, and to formulate points of departure that may eventually lead to the development of classification system for bird behaviour recognition. This goal is attained done by studying what knowledge can be derived from descriptive statistics and the learning algorithm regarding good features for specific classes. Furthermore, methods for segmentation are explored.

The next chapter goes further into the approach and implementation of the techniques used for answering the main question.

# Chapter 2

# Approach and Implementation

The approach for addressing the goal of the thesis can be divided into roughly two parts. Firstly, there is the domain to which the classification task will be applied, as discussed in section 2.1. Here, some specific details on data collection and annotation are provided. Secondly, the classification stage should increase insight in what features, classification method and segmentation algorithm leads to optimal results given the data. The approach for that aspect of the thesis is given in section 2.2.

## 2.1 Domain

Since the problem domain is the specific case of classifying bird behaviour on the basis of three-dimensional acceleration data, the classifier has to deal with both the specific sequential data format and the desired domain-specific features.

### 2.1.1 Accelerometer data

The data that will be classified, concerns sequential numeric data acquired from tri-axial accelerometers. This means that the data are three-dimensional time series, in which the dimensions correspond to acceleration in *surge*(x), *sway*(y) and *heave*(z) dimension.

The data is acquired from data loggers designed in the context of the UvA-BiTS (UvA-Bird Tracking System) project, which aims at studying among other things migration, foraging and navigation strategies of birds. The loggers are attached to the back of the bird, and since it is small and light (under 20 grams) birds are not hindered in exhibiting their normal behaviours. The devices are equipped with GPS, sensors for measuring acceleration (at a sampling rate of 20Hz), temperature, and air pressure. A solar panel avoids the battery becomes depleted. Measured data is stored on a 4MB flash memory, while transmission occurs using a Zigbee radio transceiver. This latter features makes it possible to download data from and upload new settings to the device using (portable) base stations.

The intervals at which accelerometer data is measured and stored depends on the settings defined by the researcher on the one hand, and whether the hardware is able of doing measurements. Depending on the settings, no or less samples are taken if the battery status is low and if no GPS-signal is present. Under ideal circumstances, the device is capable of taking 10 seconds (200 samples) of accelerometer data, with a 3-second interval for doing a new gps fix. However, for reason of the high power consumption, this setting is rarely used if the device is attached to a bird.

### 2.1.2 Annotating accelerometer data

Preceding work on classification of birds (e.g. Bom (2010)) relied on real-time labelled data. It has been also hypothesised that this way of annotating data results in noise, which possibly reduces accuracy in both training and classification (Hueting et al. (2011),Droeppelmann et al. (2011), de Bakker (2011)). Thus, two general drawbacks can be identified. Firstly, the annotations may be noisy due to limitations in visibility and the precision of the timing (teacher noise). Secondly, annotations cannot easily be revised afterwards. In order to overcome these problems, biologists have developed a different approach for assigning typical behaviours to accelerometer data.

This approach consists of making videos of birds equipped with the data loggers, which allows for annotating behaviours afterwards. Although there may arise some 'teacher noise', it is to be expected that annotations will be less noisy, since the researcher can replay the video and alter the labels. Further, annotations can be revised afterwards if another classification scheme is desired. Lastly, the videos and according accelerometer signals together build up to a 'library' of data, possibly useful for later application.

In order to facilitate the annotation of accelerometer time series using videos, a dedicated tool is developed in the context of this thesis. The tool, written in MATLAB, consists of a Graphical User Interface showing a loaded time series file and movie. The movie is synchronised with the relevant accelerometer data on the basis of the timestamps and sampling rates of both files. The user can scroll trough all accelerometer data (per sample, a consecutive series of measurements), while the corresponding video fragment is found. By playing the movie, the user is able to relate the data stream to particular bird movements. Subsequently, the concerned part of accelerometer data can be annotated and exported as data or as a graph.

Besides of making the annotation less time intensive and precise, the tool allows a quick visualisation of the annotating process. Furthermore, the set-up of the tool allows for future extension with analytical function. Figure 2.1 shows a screenshot of the annotation tool.

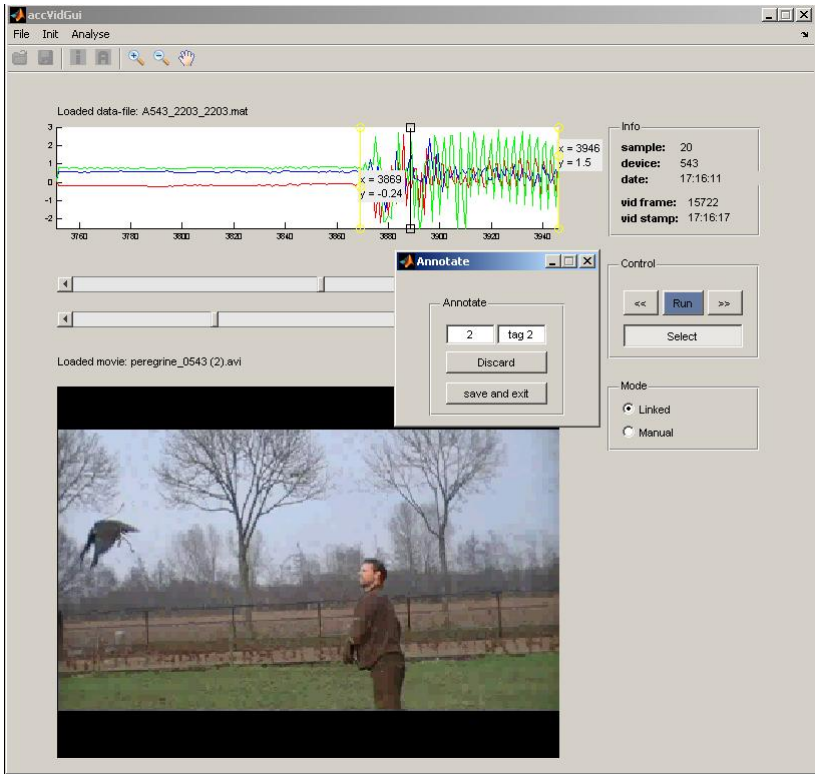



Figure 2.1: A screenshot of the annotation tool developed for the project.

### 2.1.3 Classes

Before learning the accelerometer data can occur, the data set has to be labelled with classes. On the one hand, the classes have to be relevant from the perspective of biological or ecological research. On the other hand, classes should correspond with the data in such a way a discriminative model can be trained. The relation between these two aspects is usually not obvious, and it is therefore a good idea to experiment with a broad range of classes and models. However, the intuition is that from a machine learning perspective it is usually a better idea to define low-order, easily recognisable, classes rather than high-level behaviours. The former type can be afterwards combined to interesting behavioural patterns using context information, while vice versa is not possible.

The labelled data sets have been delivered by a biology student working on a related project. Two of these data sets are used in this project, the first concerning data of a peregrine and the second set contains labelled data of gulls. However, the first data set is used for experimenting with different methods, while the second, larger data set is used for the reporting.

There are some limitations to the use of the available data. The annotations cover a wide range of behaviours, and it is expected that not all these behaviours can be classified using merely accelerometer data. Furthermore, some classes

only occur for a very short period, which makes it unfeasible to include these to the classification process. Furthermore, there are actions that compose behaviour, for example pulling, picking are part of eating. Those actions normally are very short in duration and may be difficult to recognise using fixed window feature extraction. For these reasons, some classes are excluded from the classification list. In order to reduce the skewness, the data in some classes is downsampled. A complete list of the annotated behaviours can be found in table A.2 in the Appendix.

## 2.2 Approach

The preceding section has introduced the domain of the accelerometer data, the classes that are selected and the way the labelled data set is constructed. This section then, goes further into the automatic classification process. Three aspects are discussed, namely feature extraction, learning algorithm and evaluation, and segmentation of new time series.

### 2.2.1 Data preprocessing

The data preprocessing stage consists generally of cleaning the data and representing the data in a manner that is optimal for learning algorithm. The next two sections describe how in this project is dealt with preprocessing.

#### Feature extraction

Features are extracted from the data set that contains signals $A$ of individual classes. The window $W$ is slided over the data, using some overlap for making the features less dependent on the start and end points:

$$\overrightarrow{W}_i = \{A_d^{t_i}, ..., A_d^{t_i+n}\} \text{ with } d = \{x, y, z\}, n = windowsize \text{ and } t_{i+1} = t_i + overlap$$

The time series are not directly input to the classifier, but the three-dimensional signals are represented by a feature vector including the class label $C$ of some window $W$:

$$\overrightarrow{F}_W = \{C, f_1, ..., f_n\} \text{ where } f_i \text{ is a feature.}$$

The window size can be of influence on some features, and earlier experiments have shown that a window size of 20 frames (1 second), with an overlap of 10 frames works well on this data. Further, in order to level out high-frequency noise, a fourth order Butterworth low-pass filtered is applied to the windows before features are extracted. Using this filter slightly improved accuracy in own experiments. Finally, each window is converted to a feature vector, which is used as the training and validation input for the classifier.

Distance based learning algorithms, such as Support Vector Machines, require normalised feature values. Algorithms that make decisions on for example information gain measures, such as decision trees, do not have this requirement. Normalisation of the features thus will be applied depending on the classifier used in the later experiments.

The complete feature vector contains 46 features from the time, frequency and time-frequency domain. The selection of the features is based on insights of the raw signals on the one hand, and on features used in preceding studies (Preece et al. (2009), Gyllensten (2010)) on the other hand. All calculations are programmed in MATLAB. A complete list of features can be found in table A.1 in the appendix, while the most important details of the features are discussed below.

**Features in the time domain** Some information can be derived directly from the raw signal in the time domain. Mostly, these features are simple descriptive statistics, such as the mean and standard deviation. However also the mean and standard deviation of the pitch and roll are calculated since these measures inform about the position of the bird (for convenience, the radial values are converted to degrees):

$$pitch = \arctan(\frac{x}{\sqrt{y^2+z^2}})$$

$$roll = \arctan(\frac{y}{\sqrt{x^2+z^2}})$$

Furthermore, the trend of the pitch and roll within a window can be of interest for some behaviours. These features are calculated by fitting a linear function on the smoothed (moving average) pitch and roll values. The first coefficient represents the linear trend of the pitch and roll.

Some signals may show correlation between the different dimensions over time. The correlation between the dimensions $x.y, x.z, y.z$ is calculated:

$$correlation(x,y) = \frac{\sum_i^n x_i yi - n*mean(x_i)mean(y_i)}{(n-1)std(x_i)std(y_i)} \text{ where } i \text{ is frame number}$$

Lastly, a feature for measuring motionlessness (adapted from Gyllensten (2010)), might discriminate between active and still behaviours. The motionlessness is calculated by measuring the number of frames that do not vary according to a threshold value.

**Features in the frequency domain** The simple descriptive features pertain only to the signal in the time domain. However, a classifier may be able to distinguish between frequency and amplitude components of a signal as well. The frequency components can be found by means of discrete Fourier analysis, and the implementation used is the Fast Fourier Transform:

$$F(x) = |\sum_i^{n-1} x_i e^{\frac{-2\pi j}{n} ik}|_k^2 \text{ for } k = 1, ..., 10$$

In order to avoid 'spectral leakage' in non-periodic signals, a Hamming window is applied to the selected frames before doing the Fourier transform. Furthermore, the first DC component is excluded from the periodogram.

From the Fourier components, the maximum amplitude is extracted, as well as the corresponding frequency. Furthermore the energy in four subbands should give an idea what frequency ranges are important in the signal. Lastly, the complexity of the signal is calculated using the entropy measure:

$$entropy = \sum_k F(x)_k log(F(x)_k)$$

**Features in the time-frequency domain** The Fourier features do not provide information about where in the time series particular frequencies occur. In order to combine time and frequency information, wavelet decomposition can be used. In most studies discussed in Preece et al. (2009), the detail coefficients of a Daubechies wavelet mother is selected, although other wavelets have been applied too. For every decomposition level (scale) 1 to 3, simply the norm of the coefficients is calculated:

$$magnitude = |cD_1|_1, |cD2|_1, |cD3|_1 \text{ where } cA \text{ is the approximate coefficient and } cD \text{ the detail coefficient.}$$

Although the wavelet transform is closely related to Fourier analysis, it is possible that these features may be used for separating irregular signals.

#### Descriptive statistics of features

The large number of features extracted from each window raises the question whether features are correlated and to what extent the mean and deviation the features differ among the different classes. Some classifiers actively make use of such statistical measures, while decision trees deploy a heuristic approach. Nevertheless, from the point of view of understanding the relation between signal and behaviour, some knowledge about the feature distributions can be valuable.

The mean and standard deviation of the extracted features for each category in the *gull* data set are calculated. Further, in order to find dependencies between features, the cross-correlation coefficients between the features for the complete *gull* data set is calculated.

### 2.2.2 Classification

The preprocessing stage has prepared the data for training by a classifier. Also training and classification can be done in several ways, and the methods used in this thesis are reported in the following section.

#### Learning algorithm

For training and classification a Decision Tree learner is used. Decision tree algorithms are supervised, nonparametric methods for classification or regression, which means that the one and the same model is used for all instances in the train and test phase. Furthermore, no particular parametric distribution of the input data is used in constructing the tree. (Alpaydin, 2010)

Rather, nodes are added recursively using an impurity measure. This impurity measure determines whether after a (binary) split at node $m$, all instances $N_m$ following a particular branch of the split belong to the same class $C_i$, thus estimating probability of class $C_i$:

$$\hat{\mathrm{P}}(C_i|x, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$

If this probability $p_m^i$ is either 0 or 1 for all instances, the node is said to be pure, that is to say, a leaf node is reached. There are several methods to calculate the impurity of a node, and the measure used in this thesis is the entropy:

$$\exists_m = -\sum_{i=1}^{K} p_m^i log_2 p_m^i$$

One complication is that the features used in this thesis are all numeric, which implies that many splits could be made. This is usually accounted for by testing the impurity iteratively at some point and check whether the adjacent points belong to different classes. A final consideration is that trees tend to overfit the input data since minimising entropy may result in repeatedly making new splits. Therefore, splitting is ended if $\exists < \theta$. Furthermore, pruning, which stopping the splitting if the number of instances reaching a node is small, is applied.(Alpaydin, 2010)

Decision learners have some advantages compared to other machine learning algorithms. Firstly, the decision tree is a relatively simple algorithm to implement and to interpret. Particularly, the visualisation of the tree allows often for a better understanding of the decisions made by the classifier. Furthermore, decision trees usually perform well, even compared to more complicated algorithm such as Support Vector Machines. This is also true for the features extracted from accelerometer data (Gyllensten, 2010))

As a comparison, a next to a tree learner, a Support Vector Machine is trained in the experiments. Support Vector Machines (SVM) are an instance of a wider range of 'kernel machines'. The idea of the SVM is to maximise the margin between classes. This comes down to finding an optimal separability of the classes, which is hold by the 'support vectors' (weight vector). Basically, the separation is linear, and the name 'kernel machines' reflects the idea of transforming this linear relation to a non-linear function, increasing the VC-dimension. (Alpaydin, 2010) An often used kernel is the Radial-Basis Kernel Function. Support Vector Machines are discriminant-based, which makes them suitable for data than density-based classifiers.

The implementation of the Decision Tree and Support Vector Machine, the the MATLAB toolbox PRTools (Duin et al. (2007)) is used. The learning algorithm used in PRTools is based on the Quinlan algorithm, including several options for pruning, cross-validation and feature ranking. However, since the standard implementation lacks some of functionality needed here, some functions have been adapted and added. Among other things, the decision rules and confidence estimates are collected from the model trained by the PRTools implementation. This information is necessary for implementing the classification schemes (discussed in the next paragraph), interpreting the results, and creating the tree visualisation.

**Classification schemes** Training of the decision tree is done using multi-class classification on the one hand, and the decomposition of the problem into one-versus-all and one-versus-one classifiers on the other hand. These last two decomposition schemes are expected to deliver insight in how well the classes and features are separable. Furthermore, in unbalanced datasets decomposed classification problems may improve classification accuracy since binary classes are generally easier to separate. This advantage particularly pertains to one-

versus-one classification(Galar et al. (2011)).

However, the binary classifiers have to be recomposed in order to make them work on multiclass problems. Although there are several possibilities to attain this, in this thesis the decision is made to apply the most usual approach. For the one-versus-all scheme, each classifier simply votes for a class. The confidence of each classifier is retrieved into a confidence vector (Galar et al. (2011)):

$$R = (r_1, r_2, ..., r_m) \text{ with } r_i \in [0, 1]$$

The winner is the decision connected with the highest confidence:

$$class = argmax_{i=1,...m} R_i$$

The one-versus-one recomposition works analogously. It uses a matrix containing the confidences for each classifier in favour of the class $m$ in the row or column (Galar et al. (2011):

$$\begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \text{ with } R_{ij} \in [0, 1]$$

The winning class $m$ is the class with the highest sum of confidences, thus a weighted vote:

$$class = argmax_{i=1,...,m} \sum_{i \leq j \neq i \leq m} r_{ij}$$

**Evaluation of classification**

It is not sensible to train the classifier using the complete data set as this may result in overfitting the data. Instead, the instances are split into one-third test set and two-third train and validation set. These latter sets are used for tuning the parameters. The optimal results in the validation set are found by means of a 10-fold cross validation. The results reported concern the results of a mapping of the classification model to the separate test set.

Evaluation of the classification is done by calculating performance measures using a confusion matrix (table 2.1). The confusion matrix outputs the predicted classes against the estimated classes and in this way the confusion matrix provides insight in how the classifier discriminates between the different classes.

| *classified* → *true* ↓ | **class** | **other classes** |
|---|---|---|
| **class** | true positive | false negative |
| **other classes** | false positive | true negative |

Table 2.1: Confusion matrix (Alpaydin (2010))

The overall success of the classification process is usually measured simply providing the accuracy, which is the ratio of the correctly and the incorrectly classified instances:

$$Accuracy = \frac{tp+tn}{N}$$

If the distribution of instances is skewed, accuracy usually does not reflect the true performance of the classifier for all classes. Therefore, precision, recall and F are reported as well. Precision concerns what share of the positive classifications is correctly classified (relevant):

$$Precision = \frac{tp}{tp+fn}$$

Recall measures the share of true positive classes versus the positive classifications (what share of the relevant instances is retrieved):

$$Recall = \frac{tp}{tp+fp}$$

Lastly, the F-score is a weighted measure based on the precision and recall.

$$F = 2 * \frac{precision*recall}{precision+recall}$$

### 2.2.3 Segmentation

An approach of segmenting data in order to avoid misclassification in situations of signal transitions would be very welcome. In order to study the feasibility of doing this, three pilot studies are performed. Firstly, the earlier described algorithms of Himberg et al. (2001), using vector quantization, and Linaker and Niklasson (2000) using error (variance) minimisation, were implemented. Although they perform quite well on simple test data, tests on real examples from the bird data sets yield unsatisfactory results. Another drawback is that there are several parameters involved, which means that there is a lot of experimentation needed. Thirdly, inspired by a method of Nyan et al. (2006), a pilot using wavelet analysis for finding positions in the signal where strong frequency changes occur, shows several misses in segmentation. A possible explanation is that the wavelet features look for sharp changes in frequencies, while not all behaviours are expected to be separable by means of merely frequency. Thus, none of these three simple unsupervised methods for segmenting the data proves to be preferred to a supervised windowing method. Therefore, it is opted for trying another simple algorithm for finding and classifying transitions in unseen data.

The idea of this algorithm is to find a best 'fit' of a class to a particular part of the signal. Firstly, a restricted area, a window, of the input signal is selected, and slided with an overlap of one frame over the signal. For each window, feature are extracted and classified, for example according the 1-versus-all scheme. The output classes are not considered, but instead the confidence of the classifier for the best class C in window W is stored in a vector:

$$CV_w = conf_i(W), \text{ where } conf_i \text{ is the confidence of winning class } i$$

If there is a transition from one class to another within the window, it is expected that the confidence for that window decreases. Therefore, if such a drop in the confidence vector is found, a possible breakpoint is set to the centre of the window. If a low confidence is stable, that is, if it holds on for a number of frames, or if the difference between the classifiers is small, this part of the signal might be new or difficult to classify. This is an extension of the algorithm,

using the ideas of Linaker and Niklasson (2000) for detecting stable patterns. If the splits are made, the segments between the breakpoints are concatenated, and classified according to the training schema. The pseudecode of the basic algorithm can be found in 1.

**Algorithm 1** Segmentation algorithm

```
  Function SegmentTS
Require: timeseries, windowsize, shift, δ, ϵ, classifier
  C = [], V = [], S = []
  for i = timeseries(1) to timerseries(end)-windowsize do
    w_i ← timeseries(i)totimeseries(i + windowsize)
    ft_w ← Extractfeatures
    Get confidences according to classification scheme
    V_i ← confidences
  end for
  Va ← MovingAverage(V)
  for i = 1 : length(V) do
    dV ← calculateDistance(Va, V)
    C ← δLinakeretal.(2000)
    if dV ≥ ϵ then
      S_i ← breakpoint
    end if
  end for
```

The expectation is however, that very segments windows will result in a poor classification and therefore such segments will be often falsely labelled as uncertain.

# Chapter 3

# Results

In the preceding chapter, the decisions made regarding the classification and segmentation process have been explicated. This chapter discusses the main results of the proposed methods on a data set. Firstly, the set-up of the experiments is provided, while the main part of this chapter is devoted to the interpretation of the results.

## 3.1 Experimental design

The classification experiments are done using the *gull* data set. The data set is preprocessed according to the method described in section 2. The selected classes and total duration of the behaviours are listed in table 3.1. In order to reduce the skewness of the data, the number of training examples for *stand* is reduced. Still, considerable lower amounts of data is available for the classes *fight* and *walk*, which may impact classification negatively.

| **Nr** | **Class** | **measurements** |
|---|---|---|
| 1 | stand | 1664 |
| 2 | flap | 193 |
| 3 | soar | 46 |
| 4 | walk | 193 |
| 5 | fight | 11 |
| 6 | clean fur (care) | 14 |
| 7 | brood | 173 |

Table 3.1: Behaviours and actions of the peregrine selected for classification, the total time in seconds for each class

The different analyses corresponding with the next sections are described below.

**Analysis of the features** Firstly, the results of the simple descriptive statistics are discussed, including the possible implications for the classification process (section 3.2.

**Multiclass classification** Classification results of the gull is discussed briefly. As a classifier comparison, the multiclass experiment is repeated using a

Support Vector Machine algorithm, which uses a 1-versus-all classification schema. (section 3.3)

**Binary classification** Results of the two binary classification schemes, 1-versus-all and 1-versus-1, on the *gull* data set are discussed. There is attention for the difference between these schemes and the multiclass version, and for the decision trees of the difficult classes. (section 3.4.1 and 3.4.2)

**Segmentation** Results of the experiments on finding classes in unseen data using the one-versus-one scheme and using the algorithm as described in section 2.2. (section 3.5)

## 3.2 Analysis of the feature set

Before classifying the instances, it is interesting to have some attention for the statistical properties of their features. In order to find the correlations between individual features, cross correlations between the features in the *gull* data set are calculated, which can be found in Appendix C. Also plots of the mean and standard deviations of the feature vectors of each class can be found Appendix B. The most interesting observations are discussed in the following paragraphs.

**Correlations between features** Especially in large feature sets, it is well possible that there are correlations between individual features. These correlation can be of influence on which feature the classifier chooses in making its decision. Furthermore, strong correlations may indicate redundancy in the feature vectors.[1]

The correlation table C shows some correlation that might be expected, for example between the mean values for $x$ and $y$ on the one side, and roll and pitch on the side. However, there also more surprising results. The wavelet features tend to correlate strongly with the standard deviation (pitch,roll,$x$,$y$,$z$), while this is not true for the Fourier features. Also the motionlessness features correlates (however negatively) with the standard deviation, which is not that surprising given that both features variation in the signal. The mean values for the $z$ and $y$ axis show a strong negative dependency.

The Fourier features generally have rather low correlation coefficients although there is some variation between the axis. Particularly the dominant frequency for the $x$ and $z$ dimension have hardly any relation with other features. Likewise, the trend of the pitch is almost uncorrelated and the same goes for the correlation between the axis $x - y$, and $y - z$.

**Mean and standard deviation** Some behaviours can be directly distinguished from others on the basis of the simple features. For example, the mean pitch is moderately negative (the devices is tilted backwards) if the birds stands or walks, while if flapping or fighting, the pitch is higher. This is also true for

[1] The coefficient table is inspected manually without further calculations. If *strong correlation* is mentioned, a threshold of $|0.8|$ is followed. For *weak correlation* values lower than $|0.3|$ are considered.

class *brood*, in which the bird sits stretched on the eggs. Notable are different values for the roll, of which one would think that this feature should average around zero. In the active behaviours however, there seems the tendency to be rolled to left. This might be due to offsets in the accelerometer. For the other 'simple' features, it is difficult to find clear differences between the classes. It is interesting to see however, that the standard deviation of brooding is for almost all features around zero.

A look at the frequency and time-frequency domains reveals that a distinction can be easily made between the *stand* and *brood* on the one side, and the more active patterns on the other hand. This can be seen by the amount of energy in the signal and the higher dominant frequencies for the active patterns. The difference between brood and stand is however more difficult to made using these features. Further, the wavelet features show a high standard deviation, and generally high values for the higher scales in the heave (z) dimension.

## 3.3 Multiclass classification

In the multiclass experiments, a single decision tree is trained on all classes. Training is done on two-third of the total set of labelled data, and cross-validation is used in order to find the optimal parameters. The model is then tested on the rest two-third instances in the test set. The best cross-validation results (accuracy of 0.91) have been reached with forward pruning.

The confusion matrix of classification on the test set can be found in table 3.2. The performance measure are listed in 3.3.

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **stand** | 167 | 0 | 0 | 2 | 0 | 0 | 3 |
| **flap** | 0 | 124 | 1 | 1 | 1 | 0 | 0 |
| **soar** | 0 | 2 | 21 | 4 | 0 | 0 | 1 |
| **walk** | 3 | 6 | 0 | 121 | 0 | 2 | 5 |
| **fight** | 0 | 5 | 1 | 0 | 0 | 0 | 0 |
| **care** | 0 | 4 | 0 | 4 | 0 | 0 | 0 |
| **brood** | 0 | 0 | 0 | 2 | 0 | 0 | 115 |

Table 3.2: Confusion matrix experiment 1, multiclass

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **Precision** | 0.97 | 0.98 | 0.75 | 0.88 | 0 | 0 | 0.98 |
| **Recall** | 0.98 | 0.87 | 0.91 | 0.90 | 0 | 0 | 0.93 |
| **F** | 0.98 | 0.93 | 0.82 | 0.89 | - | - | 0.95 |
| **accuracy** | 0.92 | | | | | | |

Table 3.3: Performance measures multiclass experiment

It becomes directly apparent that the classes *fight* and *care* are alway confused with other classes. The classes *stand* and *brood* are very well separable, which might be different from what one would expect beforehand. Behaviours

*flap*, *soar* and *walk* are also quite well distinguished by the classifier, although particularly walk is incidentally confused with other classes.

The pruned decision tree of the trained classifier can be found in figure D.1 in the appendix. In general, the first decisions are made on the basis of the frequency and time-frequency features. At the lower nodes in the three, the 'simpler' features make the decisions more specific. Decisions for *fight* and *care* are mainly made on the basis of the pitch feature, and this may be the reason for the poor recognition of these classes. These classes have also the smallest amount of samples, and this probably influences the tree.

**Support Vector Machine** The results of the classification may vary between algorithms. Therefore, it interesting to consider the results if the same experiment is performed using a Support Vector Machine. Support Vector Machine have proved yield good results in preceding experiments. Nevertheless, results vary considerably according to the kernel parameter $C$ and $\gamma$, and the results reported here are those with the best parameter settings. Furthermore, features are normalised of the features is necessary before training and classification. Table 3.4 shows that, using optimal kernel parameters, the Support

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **Precision** | 0.97 | 0.98 | 0.96 | 0.96 | 0.67 | 0.38 | 1 |
| **Recall** | 0.99 | 0.96 | 0.93 | 0.93 | 0.80 | 0.75 | 0.99 |
| **F** | 0.98 | 0.97 | 0.95 | 0.94 | 0.73 | 0.50 | 1 |
| **accuracy** | 0.96 | | | | | | |

Table 3.4: Performance measures multiclass experiment with Support Vector Machine

Vector Machine performs slightly better than the multiclass decision tree. In particular, recognition of the classes *care* and *fight* has been improved. This improvement can be probably devoted to the fact that Support Vector Machines are, compared to decision trees, less sensitive for skew instance distributions. Furthermore, the Support Vector Machine algorithm is based on the 1-versus-all scheme, while the decision tree in this section is a multiclass classifier.

Although the results of the Support Vector Machines are better than the Decision Tree outcomes, the following analyses are based on Decision Tree classifiers. This is, as explicated before, because the tree algorithm is better suited for the kind of analysis done in this thesis.

## 3.4 Binary classification

The poor classification of the classes *care* and *fight* using multiclass decision trees, raises the question to what extent these classes can be better separated using a binary classification scheme. The next two sections deal with the 1-versus-all scheme and the 1-versus-1 scheme respectively.

### 3.4.1 1 versus All classification

The 1-versus-All classifier is made by classifying one class as the positive class and train it against all other classes together. In this case thus, seven separate binary classifiers are made, and recomposed according to the method described under section 2.2. The recomposition is used to classify the test set, and the resulting confusion matrix of this test is found in 3.5

The two problematic classes are slightly better classified, which is reflected in the higher F-value of these classes (table 3.6). Overall, the performance has increased slightly compared to the multiclassifier.

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **stand** | 169 | 0 | 0 | 2 | 0 | 0 | 1 |
| **flap** | 0 | 118 | 1 | 8 | 0 | 0 | 0 |
| **soar** | 0 | 1 | 24 | 0 | 0 | 0 | 3 |
| **walk** | 1 | 4 | 0 | 126 | 0 | 5 | 1 |
| **fight** | 0 | 0 | 2 | 2 | 2 | 0 | 0 |
| **care** | 0 | 1 | 0 | 4 | 0 | 3 | 0 |
| **brood** | 0 | 0 | 0 | 0 | 0 | 0 | 117 |

Table 3.5: Confusion matrix 1 versus all

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **Precision** | 0.98 | 0.93 | 0.86 | 0.92 | 0.33 | 0.38 | 1 |
| **Recall** | 0.99 | 0.95 | 0.89 | 0.89 | 1 | 0.38 | 0.96 |
| **F** | 0.99 | 0.94 | 0.87 | 0.90 | 0.5 | 0.38 | 0.98 |
| **accuracy** | 0.94 | | | | | | |

Table 3.6: Performance measures 1 versus all

A look at the decision trees of the seven classifiers may say more about on which basis the categories are separated from other instances. Considering figure 3.1, it is interesting to find that the fight class would be recognised by relatively high energy levels in the higher frequency domain. Furthermore, a high standard deviation for the roll and pitch seems to correspond with the idea that in a fight the bird has strong body movements. Looking at figure 3.2 then, almost the same story applies. The relatively high energies and standard deviations for roll and pitch imply often complex signals. Indeed, the classes *fight* and *care* are most often confused with classes that are characterised by such signals as well, such as *walking*.

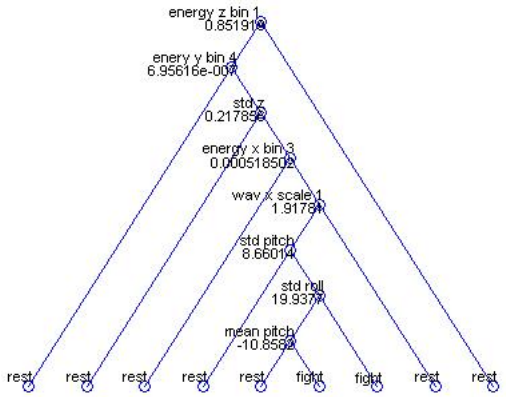



Figure 3.1: Class *fight* versus *all*

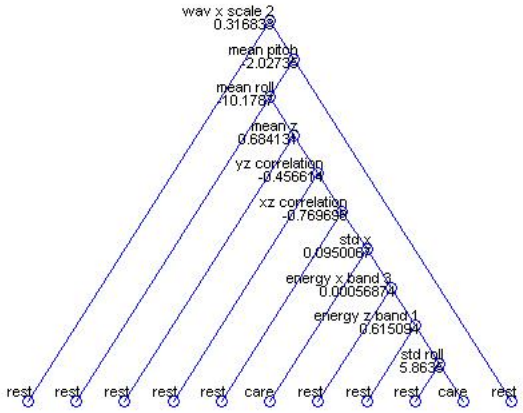



Figure 3.2: Class *Care* versus *all*

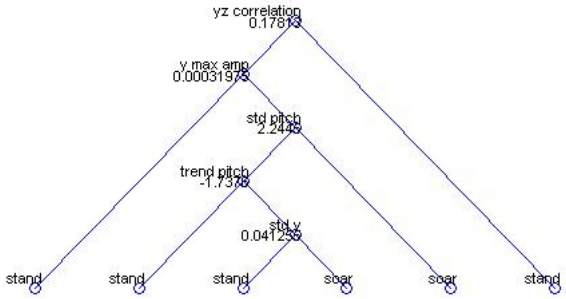



Figure 3.3: Class *stand* versus *soar*

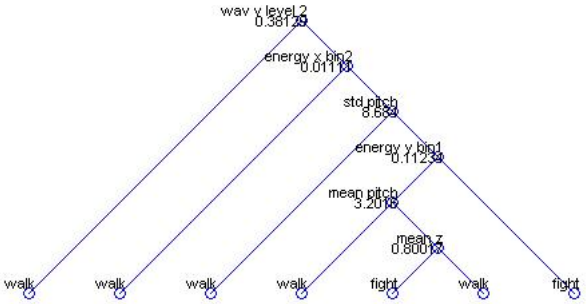



Figure 3.4: Class *walk* versus *fight*

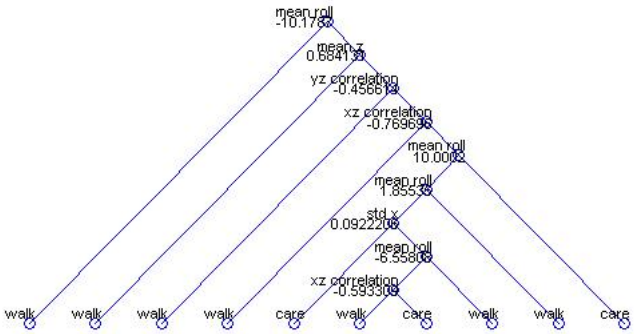



Figure 3.5: Class *walk* versus *care*

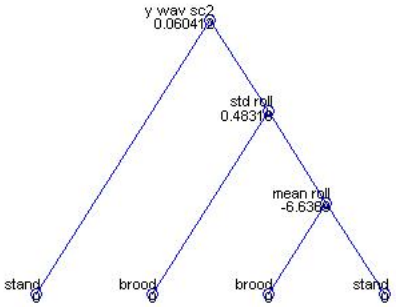



Figure 3.6: Decision tree *stand* against *brood*.

### 3.4.2 1 versus 1 classification

The 1-versus-1 scheme produces classifiers for each possible pair of categories, namely 21 in this instance. These classifiers are recomposed following the method described in section 2.2.

The confusion matrix (table 3.7) and the evaluation measures (table 3.8) show a minimal increase in the accuracy, although this is not on the account of the difficult classes *care* and *fight*. Thus, all classes slightly gain from the one-versus-one classifier, except for the problematic classes *fight* and *care*. This implies that those classes are generally difficult to separate from any other class in the data set.

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **stand** | 169 | 0 | 0 | 2 | 0 | 0 | 1 |
| **flap** | 0 | 123 | 0 | 4 | 0 | 0 | 0 |
| **soar** | 0 | 3 | 24 | 1 | 0 | 0 | 3 |
| **walk** | 2 | 3 | 0 | 128 | 0 | 2 | 1 |
| **fight** | 0 | 1 | 1 | 1 | 2 | 1 | 0 |
| **care** | 0 | 1 | 0 | 5 | 0 | 2 | 0 |
| **brood** | 0 | 0 | 0 | 0 | 0 | 0 | 117 |

Table 3.7: Confusion matrix 1 versus 1

| | **stand** | **flap** | **soar** | **walk** | **fight** | **care** | **brood** |
|---|---|---|---|---|---|---|---|
| **Precision** | 0.98 | 0.97 | 0.86 | 0.93 | 0.33 | 0.25 | 1 |
| **Recall** | 0.99 | 0.93 | 0.96 | 0.91 | 1 | 0.4 | 0.98 |
| **F** | 0.99 | 0.95 | 0.91 | 0.92 | 0.5 | 0.31 | 0.99 |
| **accuracy** | 0.95 | | | | | | |

Table 3.8: Performance measures 1 versus 1

Since 21 different decision trees are constructed, only the most relevant ones are discussed here. Figures 3.4 and 3.5 show the decisions on which classification between *walk* and respectively *fight* and *care* are made. The features selected by the decision tree to separate between *walk* and *care* is solely made on the basis of the simple features, such as body roll and the correlations. This on itself reasonable if there is little difference between these classes as far as the frequency features are concerned. On the other hand, the *care* actions are probably very irregular, implying that simple features lack the precision. With regard to the classes *walk* and *fight* it is less clear for what reason the classification fails. Possibly, the same explanation goes here, namely that the signal is too irregular to measure feature values different enough from those of the class *walk*.

In this context, it is interesting to compare the classes *stand* and *soar*, which on first sight may be difficult to distinguish. Figure 3.3 shows that the main decision is made on the basis of the standard deviations of pitch and the y-axis. These, and the few other simple features, are apparently sufficient for classifying *stand* and *soar*.

Another interesting observation is the separation between *stand* and *brood*,

which on first sight would be expected to be difficult to separate. Figure 3.6 shows the decision tree for the corresponding classifier. It is interesting to see that here the wavelet coefficient for the $y-$dimension plays a role, for which is not a clear explanation. Furthermore, it is surprising that a decision is made using the roll features, which means that the sidewards tilt is an important discriminator. This outcome is inquired about with the biologist who annotated the data, and it corresponds with her observations.

## 3.5 Segmentation

Classification of unseen data is done by creating a randomly sampled time series from the test data in which each class is at least for one period present, and in which the durations of the behaviours vary. Algorithm 1 is applied to the sample, with different parameters for window size, $\epsilon$ and $\delta$. The shift is always set to 1. Figure 3.7 presents the results for a pilot with window size of 10 frames and $\epsilon = 0.6$. The detection of the stability (algorithm parameter $\delta$) is omitted in this pilot.

The first plot in figure 3.7 refers to the original signal with the correct transitions. The second plot concerns the confidence vector and the third plot the difference vector. Peaks in this latter plot indicate a possible transition. Although most transitions are found, there is a slight lag in the detection, which results from the maximum precision that can be obtained given the window size and the overlap. Furthermore, some classes have a longer period of low confidence, and the final classifier can output these as uncertain. In this way, this algorithm can classify on the basis of confidence, while also transitions are taken into account.

Nevertheless, the algorithm is not perfect. Firstly, it is quite expensive to slide a window with a shift of 1. Secondly, the parameters imply that some tuning is needed. A third comment is related to the classifier. The classifiers of the experiments are trained for a window size of 20 frames, while the detected segments vary in length. This results, depending on the signal, often in a worse accuracy on shorter or longer windows. Experiments have been done with training the classifiers with longer and shorter windows, and while the effects on the training instances is minimal, testing on instances with different window size decreases accuracy.

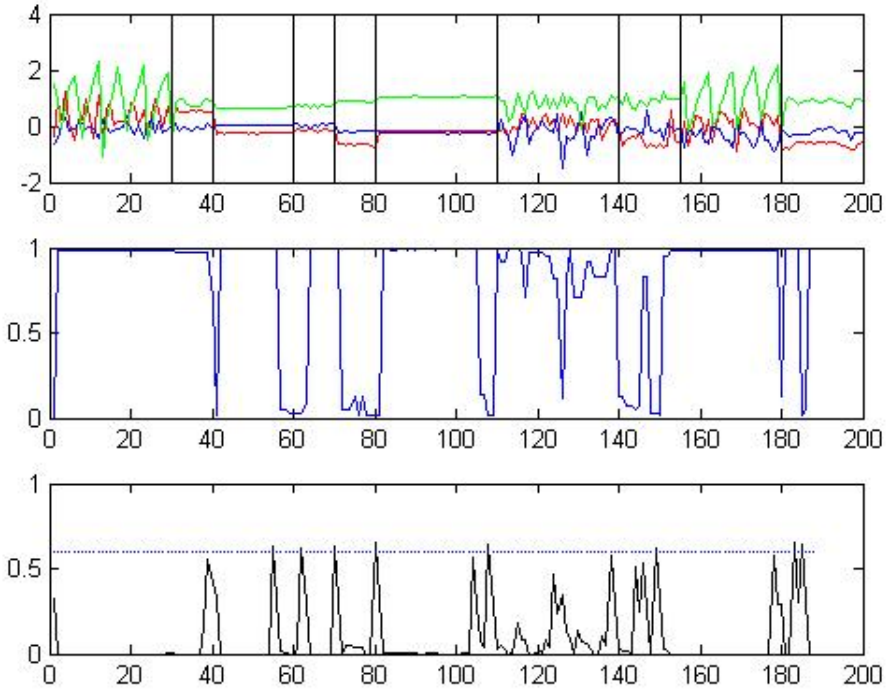



Figure 3.7: Result of a segmentation pilot. Verticals in subplot 1 are true transitions, dashed line in subplot 3 is $\epsilon$.

# Chapter 4

# Conclusion and Discussion

Classifying accelerometer data for recognition of bird behaviours can be done by means of several approaches. This writing has explored some of them regarding data annotation, feature analysis, classification and segmentation. Firstly, an annotation tool is developed in cooperation with biologists. Secondly, a wide range of features has been considered and different training schemes are applied. Further, a pilot for supervised segmentation is performed. Now, the results of these tasks end in conclusions about the data and methods, and in recommendations for further research.

## 4.1 Conclusions and discussion

With regard to the annotation of classes using video fragments, it seems that this method improves the overall classification compared to the real-life annotation. However, this improvement is not quantified since the experiments here are done using a set-up different from preceding classification experiments. This method proved to be also helpful in increasing the understanding of the relation between behaviour and signal.

On the basis of the descriptive statistics of the features and the decision tree outputs, it appears that most of the concerned behaviours are well separable by means of rather simple features. In almost all class separations, the simple features such as roll, pitch, and standard deviations are very relevant. Also the Fourier features, mainly the amplitudes in different frequency bands, appear to be helpful. The contribution of the wavelet features is difficult to interpret given the obtained results. Since the wavelet features are correlated with other features, it is problematic to assess their value with regard to the decisions. Binary classification then, appears to perform better on all classes than the multiclass variant.

The pilot with the segmentation algorithm shows that the 'supervised' approach to segmentation may be viable if the computational cost is not an issue. Compared the pilots done with the unsupervised methods, this segmentation algo-

rithm is more precise and enables excluding uncertain classifications.

In some respects, the results can be mitigated. Firstly, the relatively small data set and the skewed distribution of the class instances may have affected the classifier. The experiment with the Support Vector Machine indeed underpins this surmise. Secondly, although the correct methodological procedure is followed, the trained model might perform worse on data that is collected from other birds or birds in other circumstances. This might be true since the amount of instances trained in this project is relatively small, and therefore may contain little variation in the behaviours.

## 4.2 Discussion and recommendations

Several new questions have arisen on the basis of these results, which form interesting topics for further research. Altogether, it is hoped that these findings and recommendations have their modest contribution to the eventual development of classification system for bird behaviours on the basis of accelerometer data.

Firstly, classification of behaviours consisting of repetitive movements reach a high classification accuracy and precision. Some behaviours however, lack a clear periodicity or have irregular patterns, and these appear to be more difficult to classify using a fixed window. A reason for this is probably that the features, dependent on the window size, may level out the true irregularities. In its turn, the features resemble those of other classes too much to make a good separation. This difficulty may be addressed using smaller windows and more specific features that are independent of the window size. Possibly, wavelet features are valuable here. Since, this approach needs a flexible scheme, it is probably a good idea to do this in combination with a binary classification scheme.

Related, feature extraction has appeared to be sensitive as regards to the window size. Using different window sizes for training and testing decreases overall accuracy. One might overcome this by using solely features that do not depend on window size, but this implies that the frequency feature cannot be calculated accurately. Another interesting idea is to train instances using windows that correspond to the length of their annotations. If there is enough labelled data, this means that one will get a proper feature representation, based on the distribution of the duration of behaviours. Probably, this will improve the accuracies of flexible window classification. with the best features for those classes.

Future research may also be pointed towards studying the dependencies between features and the classifier in more depth. This is particularly interesting in the light of the correlations that exist between some features. Lastly, if using flexible windows turns out to be successful, the segmentation algorithm can be further refined since is depends largely on the performance of the classifier. Using subsets of features will decrease computational costs.

# Bibliography


Alpaydin, E. (2010). *Introduction to Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, 2nd edition.

Bao, L. and Intille, S. S. (2004). Activity recognition from user-annotated acceleration data. In Ferscha, A. and Mattern, F., editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg.

Bom, R. (2010). Can speed and tri-axial acceleration measured by biologgers be used to classify oystercatcher behaviour? Master's thesis, University of Amsterdam, The Netherlands.

de Bakker, M. P. (2011). Automatic recognition of oystercatcher behaviour using segmented time series. Technical report, University of Amsterdam, The Netherlands.

Droeppelmann, S., Gumbs, D., Smit, F., and Latour, S. (2011). Examining the oystercatcher's timebudgets using clustering on accelerometer data. Technical report, University of Amsterdam, The Netherlands.

Duin, R., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D., and Verzakov, S. (2007). Prtools4.1, a matlab toolbox for pattern recognition. Technical report, Delft University of Technology.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., and Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761 – 1776.

Guo, Y., Poulton, G., Corke, P., Bishop-Hurley, G., Wark, T., and Swain, D. (2009). Using accelerometer, high sample rate gps and magnetometer data to develop a cattle movement and behaviour model. *Ecological Modelling*, 220(17):2068 – 2075.

Gyllensten, I. (2010). Physical activity recognition in daily life using a triaxial accelerometer. Master's thesis, Royal Institute of Technology, Stockholm, Sweden.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmaki, J., and Toivonen, H. (2001). Time series segmentation for context recognition in mobile devices. In *Proceedings of the 2001 IEEE International Conference on Data Mining*.

Hueting, M., Jurriaans, R., v.d. Veen, M., and v.d. Weij, B. (2011). Classifying bird behaviour based on acceleromter data. Technical report, University of Amsterdam, The Netherlands.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '03, pages 2–11, New York, NY, USA. ACM.

Linaker, F. and Niklasson, L. (2000). Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN'00*, volume 6, pages 323–328.

Mantyla, V.-M., Mantyjarvi, J., Seppanen, T., and Tuulari, E. (2000). Hand gesture recognition of a mobile device user. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 281 –284 vol.1.

Martiskainen, P., Jarvinen, M., Skon, J.-P., Tiirikainen, J., Kolehmainen, M., and Mononen, K. (2009). Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied Animal Behaviour Science*, 119:32–28.

Nyan, M., Tay, F., Seah, K., and Sitoh, Y. (2006). Classification of gait patterns in the time-frequency domain. *Journal of Biomechanics*, 39(14):2647 – 2656.

Preece, S., Goulermas, J., Kenney, L., and Howard, D. (2009). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *Biomedical Engineering, IEEE Transactions on*, 56(3):871 –879.

Pylvänäinen, T. (2005). Accelerometer based gesture recognition using continuous hmms. In Marques, J., Pérez de la Blanca, N., and Pina, P., editors, *Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science*, pages 413–430. Springer Berlin / Heidelberg.

Sakamoto, K. Q., Sato, K., Ishizuka, M., Watanuki, Y., Takahashi, A., Daunt, F., and Wanless, S. (2009). Can ethograms be automatically generated using body acceleration data from free-ranging birds? *PLoS ONE*, 4(4):e5379.

Wilson, R., Shepard, E., and Liebsch, N. (2008). Prying into the intimate details of animal lives: use of a daily diary on animals. *Endangered Species Research*, 4:123–137.

# Appendix A

# Features and Classes

| | **Feature** |
|---|---|
| **1** | mean pitch |
| **2** | standard deviation pitch |
| **3** | mean roll |
| **4** | standard deviation roll |
| **5** | trend of pitch |
| **6** | trend of roll |
| **7** | mean x |
| **8** | mean y |
| **9** | mean z |
| **10** | std deviation x |
| **11** | std deviation y |
| **12** | std deviation z |
| **13** | xy correlation |
| **14** | yz correlation |
| **15** | xz correlation |
| **16** | motionlessness (Gyllensten (2010)) |
| **17,24,31** | max amplitude in frequency-domain (x,y,z) |
| **18,25,32** | argmax(amplitude) in frequency-domain (x,y,z) |
| **19,26,33** | Energy in 0-2.5Hz (x,y,z) |
| **20,27,34** | Energy in 2.5-5Hz (x,y,z) |
| **21,28,35** | Energy in 5-7.5Hz (x,y,z) |
| **22,29,36** | Energy in 7.5-10Hz (x,y,z) |
| **23,30,37** | entropy of the energy spectrum (x,y,z) |
| **38,39,40** | norm wavelet coefficient at scale 1,2,3 (x) |
| **41,42,43** | norm wavelet coefficient at scale 1,2,3 (y) |
| **44,45,46** | norm wavelet coefficient at scale 1,2,3 (z) |

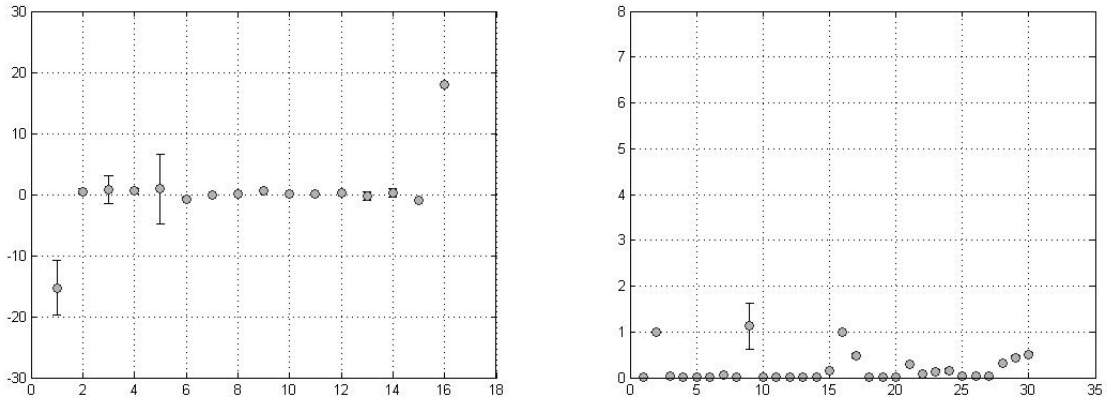
Table A.1: Legenda of features

| Nr | Class | Description |
|---|---|---|
| 1 | still (motionless) | The bird sits or stands. Motions are restricted to incidental movement of the head or wings. |
| 2 | balance | The bird sits or stands, but flaps in order to maintain balance. |
| 3 | flapping (fly) | The bird repeatedly flaps. |
| 4 | soaring (during flight) | The bird soars and does not flap. |
| 5 | eating prey | The bird chews and observes environment. |
| 6 | shaking | The birds shakes its body while still |
| 7 | still on unstable ground | The bird stands or sits on an unstable ground, for example on water. |
| 8 | moving the head | the bird moves its head. This happens for example during eating |
| 9 | observing | happens during eating |
| 10 | pull | the bird pulls on its prey |
| 11 | picking | The bird turns over. |
| 12 | walking | The bird walks |
| 13 | soaring (during landing) | The bird soars during landing |
| 14 | lifting head | The bird lifts its body, which happens during eating |
| 15 | avoid fight | The bird attempts to avoid a fight with another bird |
| 16 | fight | The bird is being attacked by another bird |
| 17 | cleaning | The bird cleans its fur |
| 18(19) | brooding | The birds sits on eggs |

Table A.2: Possible annotations in the data sets (worked out by designed by Yaiza Dronkers)

# Appendix B

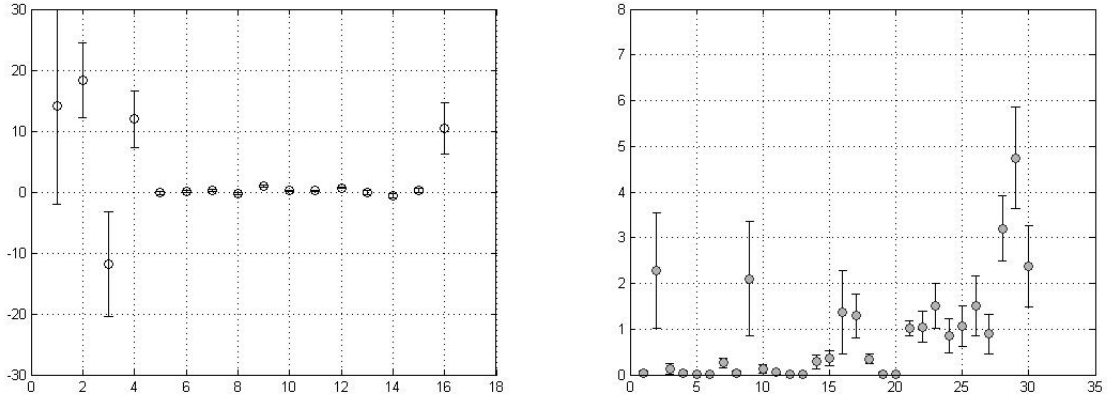
# Mean and standard deviation features




(a) stand: features 1-17

(b) stand: features 18-46

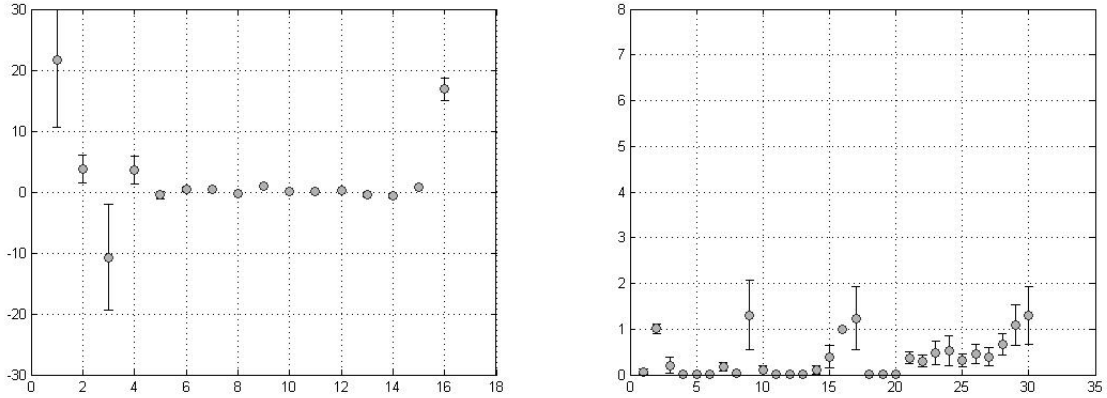
Figure B.1: Mean and standard deviation of the features for gull's behaviour *stand*



(a) flap: features 1-17

(b) flap: features 18-46

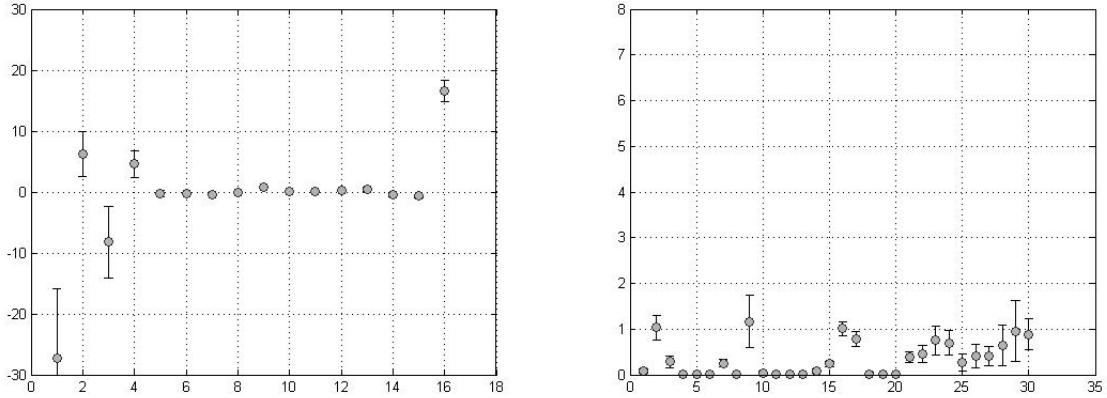
Figure B.2: Mean and standard deviation of the features for gull's behaviour *flap*



(a) soar: features 1-17

(b) soar: features 18-46

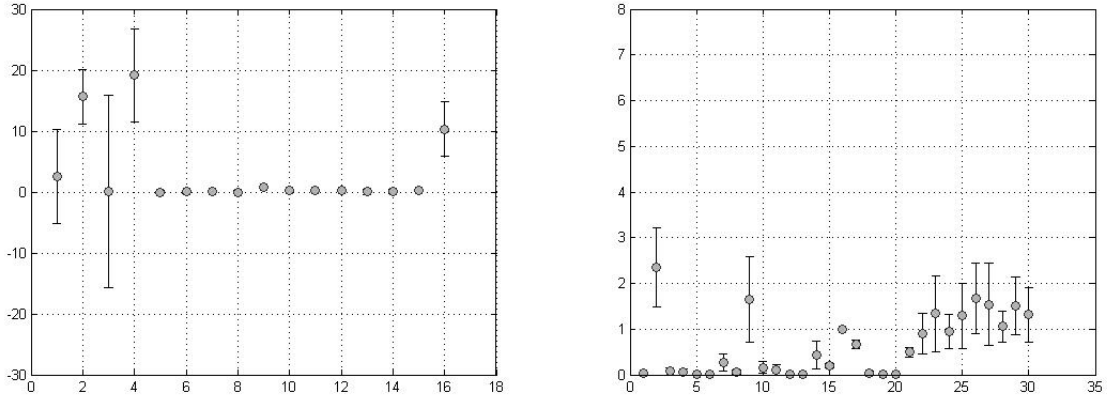
Figure B.3: Mean and standard deviation of the features for gull's behaviour *soar*



(a) walk: features 1-17

(b) walk: features 18-46

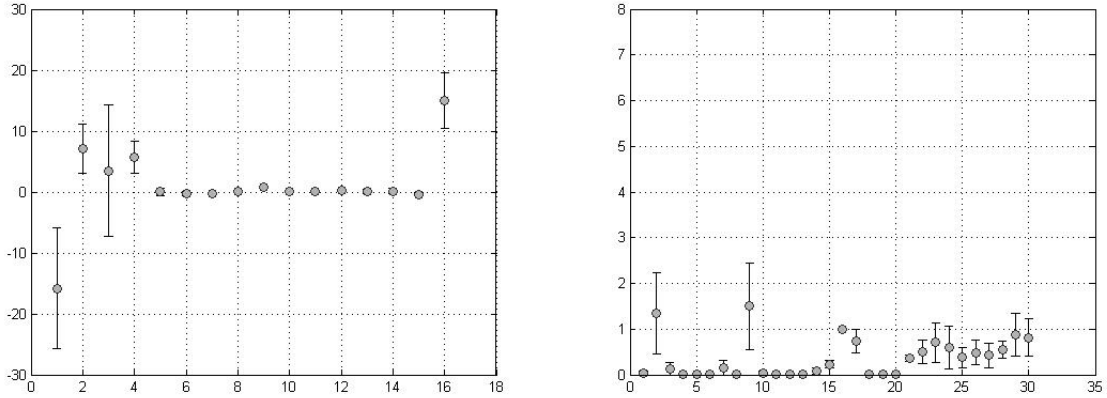
Figure B.4: Mean and standard deviation of the features for gull's behaviour *walk*



(a) fight: features 1-17

(b) fight: features 18-46

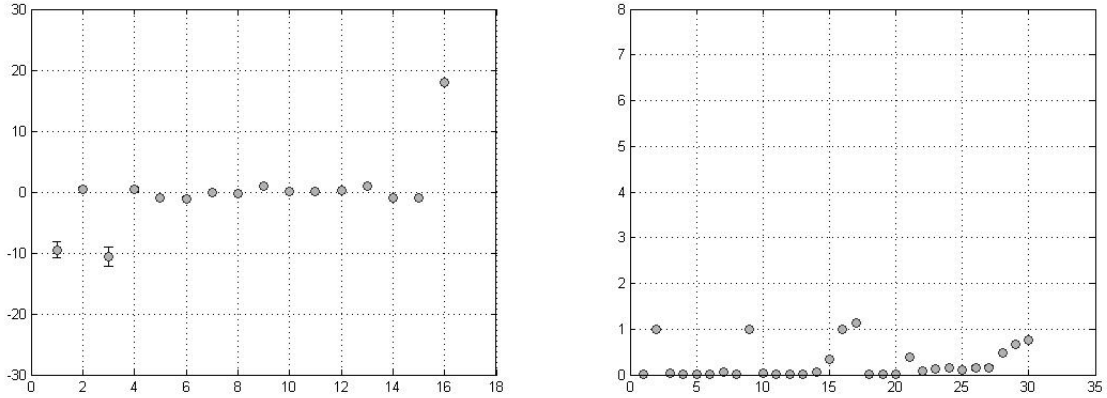
Figure B.5: Mean and standard deviation of the features for gull's behaviour *fight*



(a) clean: features 1-17

(b) clean: features 18-46

Figure B.6: Mean and standard deviation of the features for gull's behaviour *clean*



(a) brood: features 1-17

(b) brood: features 18-46

Figure B.7: Mean and standard deviation of the features for gull's behaviour *brood*

# Appendix C

# Cross correlations between features

| | 1 ptch1 | 2 ptch2 | 3 roll1 | 4 roll2 | 5 pitch3 | 6 roll3 | 7 xM | 8 yM | 9 zM | 10 xS | 11 yS | 12 zS | 13 xyC | 14 yzC | 15 xzC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1 ptch1** | 1.0 | 0.5 | 0.0 | 0.4 | 0.0 | 0.4 | 1.0 | -0.0 | 0.3 | 0.3 | 0.4 | 0.7 | -0.3 | -0.1 | 0.7 |
| **2 ptch2** | 0.5 | 1.0 | -0.2 | 0.8 | -0.0 | 0.6 | 0.5 | -0.3 | 0.2 | 0.8 | 0.8 | 0.8 | -0.2 | -0.2 | 0.7 |
| **3 roll1** | 0.0 | -0.2 | 1.0 | -0.3 | 0.2 | -0.2 | 0.0 | 1.0 | -0.7 | -0.4 | -0.5 | -0.4 | -0.3 | 0.7 | -0.3 |
| **4 roll2** | 0.4 | 0.8 | -0.3 | 1.0 | -0.0 | 0.6 | 0.4 | -0.3 | 0.2 | 0.8 | 0.9 | 0.8 | -0.2 | -0.1 | 0.7 |
| **5 pitch3** | 0.0 | -0.0 | 0.2 | -0.0 | 1.0 | -0.1 | 0.0 | 0.2 | -0.1 | -0.1 | -0.1 | -0.0 | -0.4 | 0.4 | -0.0 |
| **6 roll3** | 0.4 | 0.6 | -0.2 | 0.6 | -0.1 | 1.0 | 0.4 | -0.3 | 0.2 | 0.7 | 0.6 | 0.6 | -0.3 | -0.1 | 0.8 |
| **7 xM** | 1.0 | 0.5 | 0.0 | 0.4 | 0.0 | 0.4 | 1.0 | -0.0 | 0.2 | 0.2 | 0.4 | 0.6 | -0.4 | -0.0 | 0.7 |
| **8 yM** | -0.0 | -0.3 | 1.0 | -0.3 | 0.2 | -0.3 | -0.0 | 1.0 | -0.8 | -0.4 | -0.6 | -0.5 | -0.2 | 0.7 | -0.4 |
| **9 zM** | 0.3 | 0.2 | -0.7 | 0.2 | -0.1 | 0.2 | 0.2 | -0.8 | 1.0 | 0.3 | 0.5 | 0.6 | 0.2 | -0.6 | 0.4 |
| **10 xS** | 0.3 | 0.8 | -0.4 | 0.8 | -0.1 | 0.7 | 0.2 | -0.4 | 0.3 | 1.0 | 0.8 | 0.7 | -0.1 | -0.2 | 0.7 |
| **11 yS** | 0.4 | 0.8 | -0.5 | 0.9 | -0.1 | 0.6 | 0.4 | -0.6 | 0.5 | 0.8 | 1.0 | 0.8 | -0.1 | -0.3 | 0.7 |
| **12 zS** | 0.7 | 0.8 | -0.4 | 0.8 | -0.0 | 0.6 | 0.6 | -0.5 | 0.6 | 0.7 | 0.8 | 1.0 | -0.2 | -0.3 | 0.7 |
| **13 xyC** | -0.3 | -0.2 | -0.3 | -0.2 | -0.4 | -0.3 | -0.4 | -0.2 | 0.2 | -0.1 | -0.1 | -0.2 | 1.0 | -0.7 | -0.4 |
| **14 yzC** | -0.1 | -0.2 | 0.7 | -0.1 | 0.4 | -0.1 | -0.0 | 0.7 | -0.6 | -0.2 | -0.3 | -0.3 | -0.7 | 1.0 | -0.2 |
| **15 xzC** | 0.7 | 0.7 | -0.3 | 0.7 | -0.0 | 0.8 | 0.7 | -0.4 | 0.4 | 0.7 | 0.7 | 0.7 | -0.4 | -0.2 | 1.0 |
| **16 mot** | -0.4 | -0.8 | 0.3 | -0.8 | 0.0 | -0.5 | -0.4 | 0.4 | -0.3 | -0.8 | -0.8 | -0.8 | 0.2 | 0.2 | -0.6 |
| **17 xFA** | -0.1 | 0.3 | -0.2 | 0.3 | -0.1 | 0.4 | -0.1 | -0.2 | 0.1 | 0.6 | 0.2 | 0.2 | 0.2 | -0.1 | 0.3 |
| **18 xFF** | 0.2 | 0.5 | -0.4 | 0.6 | -0.0 | 0.3 | 0.2 | -0.5 | 0.3 | 0.5 | 0.7 | 0.6 | -0.2 | -0.2 | 0.5 |
| **19 xS1** | -0.1 | 0.2 | -0.1 | 0.2 | -0.1 | 0.4 | -0.2 | -0.1 | 0.1 | 0.6 | 0.1 | 0.1 | 0.2 | -0.1 | 0.2 |
| **20 xS2** | 0.3 | 0.7 | -0.3 | 0.7 | -0.0 | 0.4 | 0.3 | -0.3 | 0.2 | 0.8 | 0.7 | 0.6 | -0.1 | -0.1 | 0.5 |
| **21 xS3** | 0.2 | 0.5 | -0.1 | 0.5 | -0.0 | 0.3 | 0.2 | -0.2 | 0.2 | 0.5 | 0.5 | 0.5 | -0.1 | -0.1 | 0.3 |
| **22 xS4** | 0.4 | 0.7 | -0.2 | 0.6 | -0.0 | 0.5 | 0.4 | -0.2 | 0.2 | 0.6 | 0.6 | 0.6 | -0.1 | -0.1 | 0.5 |
| **23 xE** | 0.2 | 0.7 | -0.3 | 0.6 | -0.1 | 0.6 | 0.1 | -0.3 | 0.2 | 0.9 | 0.6 | 0.5 | 0.0 | -0.2 | 0.5 |
| **24 yFA** | 0.2 | 0.4 | -0.6 | 0.6 | -0.1 | 0.5 | 0.2 | -0.7 | 0.6 | 0.5 | 0.8 | 0.6 | -0.1 | -0.4 | 0.5 |
| **25 yFF** | 0.5 | 0.5 | 0.1 | 0.5 | -0.0 | 0.4 | 0.5 | 0.0 | 0.0 | 0.4 | 0.4 | 0.5 | -0.1 | -0.0 | 0.4 |
| **26 yS1** | 0.2 | 0.4 | -0.7 | 0.5 | -0.1 | 0.4 | 0.2 | -0.8 | 0.6 | 0.5 | 0.8 | 0.6 | -0.1 | -0.4 | 0.5 |
| **27 yS2** | 0.3 | 0.6 | -0.3 | 0.7 | -0.0 | 0.4 | 0.3 | -0.4 | 0.3 | 0.5 | 0.8 | 0.6 | -0.2 | -0.2 | 0.5 |
| **28 yS3** | 0.2 | 0.4 | -0.2 | 0.6 | -0.0 | 0.3 | 0.3 | -0.3 | 0.2 | 0.4 | 0.6 | 0.4 | -0.1 | -0.1 | 0.4 |
| **29 yS4** | 0.3 | 0.5 | -0.3 | 0.6 | -0.0 | 0.4 | 0.3 | -0.3 | 0.3 | 0.5 | 0.6 | 0.5 | -0.1 | -0.1 | 0.5 |
| **30 yE** | 0.3 | 0.6 | -0.5 | 0.8 | -0.1 | 0.5 | 0.3 | -0.6 | 0.5 | 0.6 | 0.9 | 0.7 | -0.1 | -0.3 | 0.6 |
| **31 zFA** | 0.3 | 0.2 | -0.6 | 0.2 | -0.1 | 0.2 | 0.3 | -0.7 | 0.9 | 0.3 | 0.5 | 0.6 | 0.1 | -0.5 | 0.4 |
| **32 zFF** | 0.4 | 0.4 | 0.1 | 0.3 | 0.0 | 0.2 | 0.3 | 0.1 | -0.0 | 0.3 | 0.2 | 0.4 | -0.0 | -0.0 | 0.2 |
| **33 zS1** | 0.4 | 0.3 | -0.6 | 0.3 | -0.1 | 0.3 | 0.4 | -0.7 | 0.9 | 0.4 | 0.6 | 0.7 | 0.1 | -0.5 | 0.5 |
| **34 zS2** | 0.6 | 0.8 | -0.3 | 0.8 | -0.0 | 0.5 | 0.6 | -0.4 | 0.3 | 0.7 | 0.7 | 0.9 | -0.2 | -0.2 | 0.7 |
| **35 zS3** | 0.4 | 0.5 | -0.2 | 0.5 | -0.0 | 0.3 | 0.4 | -0.2 | 0.3 | 0.5 | 0.5 | 0.6 | -0.1 | -0.1 | 0.4 |
| **36 zS4** | 0.4 | 0.6 | -0.4 | 0.5 | -0.0 | 0.4 | 0.4 | -0.4 | 0.4 | 0.5 | 0.6 | 0.7 | -0.1 | -0.3 | 0.5 |
| **37 zE** | 0.6 | 0.9 | -0.4 | 0.8 | -0.0 | 0.5 | 0.6 | -0.4 | 0.4 | 0.7 | 0.8 | 0.9 | -0.2 | -0.3 | 0.7 |
| **38 xW1** | 0.4 | 0.9 | -0.4 | 0.8 | -0.0 | 0.6 | 0.4 | -0.4 | 0.3 | 0.9 | 0.8 | 0.8 | -0.2 | -0.2 | 0.7 |
| **39 xW2** | 0.4 | 0.9 | -0.4 | 0.8 | -0.0 | 0.6 | 0.3 | -0.4 | 0.3 | 0.9 | 0.8 | 0.8 | -0.1 | -0.2 | 0.7 |
| **40 xW3** | 0.2 | 0.7 | -0.3 | 0.7 | -0.1 | 0.6 | 0.2 | -0.3 | 0.2 | 0.9 | 0.7 | 0.6 | -0.1 | -0.2 | 0.6 |
| **41 yW1** | 0.4 | 0.8 | -0.5 | 0.8 | -0.0 | 0.6 | 0.4 | -0.5 | 0.4 | 0.7 | 0.9 | 0.8 | -0.2 | -0.3 | 0.7 |
| **42 yW2** | 0.4 | 0.8 | -0.4 | 0.8 | -0.0 | 0.6 | 0.4 | -0.5 | 0.4 | 0.7 | 0.9 | 0.8 | -0.2 | -0.3 | 0.7 |
| **43 yW3** | 0.3 | 0.7 | -0.4 | 0.8 | -0.0 | 0.6 | 0.3 | -0.5 | 0.4 | 0.7 | 0.9 | 0.7 | -0.1 | -0.3 | 0.6 |
| **44 zW1** | 0.6 | 0.8 | -0.4 | 0.8 | -0.0 | 0.6 | 0.6 | -0.4 | 0.4 | 0.7 | 0.8 | 1.0 | -0.2 | -0.2 | 0.7 |
| **45 zW2** | 0.6 | 0.8 | -0.4 | 0.8 | -0.0 | 0.6 | 0.6 | -0.4 | 0.4 | 0.7 | 0.8 | 0.9 | -0.2 | -0.2 | 0.7 |
| **46 zW3** | 0.6 | 0.8 | -0.4 | 0.7 | -0.0 | 0.5 | 0.6 | -0.4 | 0.4 | 0.7 | 0.7 | 0.9 | -0.2 | -0.3 | 0.7 |

continued on the next page ...

| | 16 mot | 17 xFA | 18 xFF | 19 xS1 | 20 xS2 | 21 xS3 | 22 xS4 | 23 xE | 24 yFA | 25 yFF | 26 yS1 | 27 yS2 | 28 yS3 | 29 yS4 | 30 yE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1 ptch1** | -0.4 | -0.1 | 0.2 | -0.1 | 0.3 | 0.2 | 0.4 | 0.2 | 0.2 | 0.5 | 0.2 | 0.3 | 0.2 | 0.3 | 0.3 |
| **2 ptch2** | -0.8 | 0.3 | 0.5 | 0.2 | 0.7 | 0.5 | 0.7 | 0.7 | 0.4 | 0.5 | 0.4 | 0.6 | 0.4 | 0.5 | 0.6 |
| **3 roll1** | 0.3 | -0.2 | -0.4 | -0.1 | -0.3 | -0.1 | -0.2 | -0.3 | -0.6 | 0.1 | -0.7 | -0.3 | -0.2 | -0.3 | -0.5 |
| **4 roll2** | -0.8 | 0.3 | 0.6 | 0.2 | 0.7 | 0.5 | 0.6 | 0.6 | 0.6 | 0.5 | 0.5 | 0.7 | 0.6 | 0.6 | 0.8 |
| **5 pitch3** | 0.0 | -0.1 | -0.0 | -0.1 | -0.0 | -0.0 | -0.0 | -0.1 | -0.1 | -0.0 | -0.1 | -0.0 | -0.0 | -0.0 | -0.1 |
| **6 roll3** | -0.5 | 0.4 | 0.3 | 0.4 | 0.4 | 0.3 | 0.5 | 0.6 | 0.5 | 0.4 | 0.4 | 0.4 | 0.3 | 0.4 | 0.5 |
| **7 xM** | -0.4 | -0.1 | 0.2 | -0.2 | 0.3 | 0.2 | 0.4 | 0.1 | 0.2 | 0.5 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |
| **8 yM** | 0.4 | -0.2 | -0.5 | -0.1 | -0.3 | -0.2 | -0.2 | -0.3 | -0.7 | 0.0 | -0.8 | -0.4 | -0.3 | -0.3 | -0.6 |
| **9 zM** | -0.3 | 0.1 | 0.3 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.6 | 0.0 | 0.6 | 0.3 | 0.2 | 0.3 | 0.5 |
| **10 xS** | -0.8 | 0.6 | 0.5 | 0.6 | 0.8 | 0.5 | 0.6 | 0.9 | 0.5 | 0.4 | 0.5 | 0.5 | 0.4 | 0.5 | 0.6 |
| **11 yS** | -0.8 | 0.2 | 0.7 | 0.1 | 0.7 | 0.5 | 0.6 | 0.6 | 0.8 | 0.4 | 0.8 | 0.8 | 0.6 | 0.6 | 0.9 |
| **12 zS** | -0.8 | 0.2 | 0.6 | 0.1 | 0.6 | 0.5 | 0.6 | 0.5 | 0.6 | 0.5 | 0.6 | 0.6 | 0.4 | 0.5 | 0.7 |
| **13 xyC** | 0.2 | 0.2 | -0.2 | 0.2 | -0.1 | -0.1 | -0.1 | 0.0 | -0.1 | -0.1 | -0.1 | -0.2 | -0.1 | -0.1 | -0.1 |
| **14 yzC** | 0.2 | -0.1 | -0.2 | -0.1 | -0.1 | -0.1 | -0.1 | -0.2 | -0.4 | -0.0 | -0.4 | -0.2 | -0.1 | -0.1 | -0.3 |
| **15 xzC** | -0.6 | 0.3 | 0.5 | 0.2 | 0.5 | 0.3 | 0.5 | 0.5 | 0.5 | 0.4 | 0.5 | 0.5 | 0.4 | 0.5 | 0.6 |
| **16 mot** | 1.0 | -0.2 | -0.6 | -0.1 | -0.8 | -0.6 | -0.6 | -0.6 | -0.5 | -0.4 | -0.5 | -0.6 | -0.5 | -0.5 | -0.7 |
| **17 xFA** | -0.2 | 1.0 | -0.1 | 1.0 | 0.2 | 0.2 | 0.3 | 0.8 | 0.1 | 0.2 | 0.0 | 0.1 | 0.1 | 0.1 | 0.1 |
| **18 xFF** | -0.6 | -0.1 | 1.0 | -0.2 | 0.6 | 0.4 | 0.3 | 0.2 | 0.6 | 0.2 | 0.6 | 0.6 | 0.4 | 0.4 | 0.7 |
| **19 xS1** | -0.1 | 1.0 | -0.2 | 1.0 | 0.1 | 0.1 | 0.2 | 0.8 | -0.0 | 0.2 | -0.1 | -0.0 | 0.0 | 0.1 | 0.0 |
| **20 xS2** | -0.8 | 0.2 | 0.6 | 0.1 | 1.0 | 0.8 | 0.5 | 0.7 | 0.5 | 0.4 | 0.5 | 0.6 | 0.6 | 0.5 | 0.6 |
| **21 xS3** | -0.6 | 0.2 | 0.4 | 0.1 | 0.8 | 1.0 | 0.3 | 0.5 | 0.4 | 0.3 | 0.3 | 0.4 | 0.5 | 0.3 | 0.4 |
| **22 xS4** | -0.6 | 0.3 | 0.3 | 0.2 | 0.5 | 0.3 | 1.0 | 0.5 | 0.3 | 0.4 | 0.3 | 0.4 | 0.2 | 0.4 | 0.5 |
| **23 xE** | -0.6 | 0.8 | 0.2 | 0.8 | 0.7 | 0.5 | 0.5 | 1.0 | 0.3 | 0.4 | 0.3 | 0.4 | 0.4 | 0.4 | 0.4 |
| **24 yFA** | -0.5 | 0.1 | 0.6 | -0.0 | 0.5 | 0.4 | 0.3 | 0.3 | 1.0 | 0.1 | 1.0 | 0.7 | 0.5 | 0.5 | 0.9 |
| **25 yFF** | -0.4 | 0.2 | 0.2 | 0.2 | 0.4 | 0.3 | 0.4 | 0.4 | 0.1 | 1.0 | 0.0 | 0.4 | 0.3 | 0.3 | 0.3 |
| **26 yS1** | -0.5 | 0.0 | 0.6 | -0.1 | 0.5 | 0.3 | 0.3 | 0.3 | 1.0 | 0.0 | 1.0 | 0.7 | 0.4 | 0.5 | 0.9 |
| **27 yS2** | -0.6 | 0.1 | 0.6 | -0.0 | 0.6 | 0.4 | 0.4 | 0.4 | 0.7 | 0.4 | 0.7 | 1.0 | 0.8 | 0.7 | 0.9 |
| **28 yS3** | -0.5 | 0.1 | 0.4 | 0.0 | 0.6 | 0.5 | 0.2 | 0.4 | 0.5 | 0.3 | 0.4 | 0.8 | 1.0 | 0.6 | 0.7 |
| **29 yS4** | -0.5 | 0.1 | 0.4 | 0.1 | 0.5 | 0.3 | 0.4 | 0.4 | 0.5 | 0.3 | 0.5 | 0.7 | 0.6 | 1.0 | 0.6 |
| **30 yE** | -0.7 | 0.1 | 0.7 | 0.0 | 0.6 | 0.4 | 0.5 | 0.4 | 0.9 | 0.3 | 0.9 | 0.9 | 0.7 | 0.6 | 1.0 |
| **31 zFA** | -0.3 | 0.1 | 0.3 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.6 | 0.0 | 0.6 | 0.3 | 0.3 | 0.3 | 0.5 |
| **32 zFF** | -0.3 | 0.1 | 0.0 | 0.1 | 0.2 | 0.2 | 0.3 | 0.3 | 0.0 | 0.4 | -0.0 | 0.1 | 0.1 | 0.2 | 0.1 |
| **33 zS1** | -0.4 | 0.1 | 0.4 | 0.1 | 0.3 | 0.2 | 0.3 | 0.2 | 0.7 | 0.1 | 0.7 | 0.4 | 0.3 | 0.3 | 0.6 |
| **34 zS2** | -0.8 | 0.1 | 0.6 | 0.0 | 0.6 | 0.5 | 0.7 | 0.5 | 0.5 | 0.5 | 0.5 | 0.6 | 0.4 | 0.5 | 0.6 |
| **35 zS3** | -0.5 | 0.1 | 0.4 | 0.0 | 0.5 | 0.6 | 0.3 | 0.4 | 0.4 | 0.3 | 0.4 | 0.4 | 0.5 | 0.4 | 0.5 |
| **36 zS4** | -0.5 | 0.1 | 0.4 | 0.1 | 0.4 | 0.3 | 0.6 | 0.4 | 0.4 | 0.3 | 0.4 | 0.4 | 0.3 | 0.4 | 0.5 |
| **37 zE** | -0.8 | 0.1 | 0.6 | 0.1 | 0.7 | 0.5 | 0.7 | 0.5 | 0.5 | 0.5 | 0.5 | 0.6 | 0.4 | 0.5 | 0.7 |
| **38 xW1** | -0.9 | 0.4 | 0.6 | 0.3 | 0.8 | 0.7 | 0.6 | 0.8 | 0.6 | 0.5 | 0.5 | 0.6 | 0.5 | 0.5 | 0.7 |
| **39 xW2** | -0.8 | 0.4 | 0.6 | 0.3 | 0.8 | 0.6 | 0.6 | 0.8 | 0.5 | 0.4 | 0.5 | 0.6 | 0.5 | 0.5 | 0.7 |
| **40 xW3** | -0.6 | 0.5 | 0.4 | 0.5 | 0.6 | 0.4 | 0.5 | 0.8 | 0.4 | 0.3 | 0.4 | 0.4 | 0.3 | 0.4 | 0.5 |
| **41 yW1** | -0.8 | 0.2 | 0.7 | 0.1 | 0.7 | 0.5 | 0.6 | 0.5 | 0.7 | 0.4 | 0.7 | 0.8 | 0.7 | 0.6 | 0.9 |
| **42 yW2** | -0.8 | 0.2 | 0.7 | 0.1 | 0.7 | 0.5 | 0.6 | 0.5 | 0.7 | 0.4 | 0.7 | 0.8 | 0.6 | 0.6 | 0.9 |
| **43 yW3** | -0.7 | 0.3 | 0.6 | 0.2 | 0.6 | 0.4 | 0.5 | 0.6 | 0.7 | 0.4 | 0.7 | 0.7 | 0.5 | 0.6 | 0.8 |
| **44 zW1** | -0.8 | 0.1 | 0.6 | 0.1 | 0.7 | 0.5 | 0.7 | 0.5 | 0.6 | 0.5 | 0.6 | 0.6 | 0.5 | 0.5 | 0.7 |
| **45 zW2** | -0.8 | 0.2 | 0.6 | 0.1 | 0.7 | 0.5 | 0.7 | 0.5 | 0.5 | 0.5 | 0.5 | 0.6 | 0.4 | 0.5 | 0.7 |
| **46 zW3** | -0.7 | 0.2 | 0.5 | 0.1 | 0.5 | 0.3 | 0.6 | 0.5 | 0.5 | 0.4 | 0.5 | 0.5 | 0.3 | 0.5 | 0.6 |

continued on the next page ...

| | 31 zFA | 32 zFF | 33 zS1 | 34 zS2 | 35 zS3 | 36 zS4 | 37 zE | 38 xW1 | 39 xW2 | 40 xW3 | 41 yW1 | 42 yW2 | 43 yW3 | 44 zW1 | 45 zW2 | 46 zW3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1 ptch1** | 0.3 | 0.4 | 0.4 | 0.6 | 0.4 | 0.4 | 0.6 | 0.4 | 0.4 | 0.2 | 0.4 | 0.4 | 0.3 | 0.6 | 0.6 | 0.6 |
| **2 ptch2** | 0.2 | 0.4 | 0.3 | 0.8 | 0.5 | 0.6 | 0.9 | 0.9 | 0.9 | 0.7 | 0.8 | 0.8 | 0.7 | 0.8 | 0.8 | 0.8 |
| **3 roll1** | -0.6 | 0.1 | -0.6 | -0.3 | -0.2 | -0.4 | -0.4 | -0.4 | -0.4 | -0.3 | -0.5 | -0.4 | -0.4 | -0.4 | -0.4 | -0.4 |
| **4 roll2** | 0.2 | 0.3 | 0.3 | 0.8 | 0.5 | 0.5 | 0.8 | 0.8 | 0.8 | 0.7 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.7 |
| **5 pitch3** | -0.1 | 0.0 | -0.1 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.1 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
| **6 roll3** | 0.2 | 0.2 | 0.3 | 0.5 | 0.3 | 0.4 | 0.5 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.5 |
| **7 xM** | 0.3 | 0.3 | 0.4 | 0.6 | 0.4 | 0.4 | 0.6 | 0.4 | 0.3 | 0.2 | 0.4 | 0.4 | 0.3 | 0.6 | 0.6 | 0.6 |
| **8 yM** | -0.7 | 0.1 | -0.7 | -0.4 | -0.2 | -0.4 | -0.4 | -0.4 | -0.4 | -0.3 | -0.5 | -0.5 | -0.5 | -0.4 | -0.4 | -0.4 |
| **9 zM** | 0.9 | -0.0 | 0.9 | 0.3 | 0.3 | 0.4 | 0.4 | 0.3 | 0.3 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| **10 xS** | 0.3 | 0.3 | 0.4 | 0.7 | 0.5 | 0.5 | 0.7 | 0.9 | 0.9 | 0.9 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| **11 yS** | 0.5 | 0.2 | 0.6 | 0.7 | 0.5 | 0.6 | 0.8 | 0.8 | 0.8 | 0.7 | 0.9 | 0.9 | 0.9 | 0.8 | 0.8 | 0.7 |
| **12 zS** | 0.6 | 0.4 | 0.7 | 0.9 | 0.6 | 0.7 | 0.9 | 0.8 | 0.8 | 0.6 | 0.8 | 0.8 | 0.7 | 1.0 | 0.9 | 0.9 |
| **13 xyC** | 0.1 | -0.0 | 0.1 | -0.2 | -0.1 | -0.1 | -0.2 | -0.2 | -0.1 | -0.1 | -0.2 | -0.2 | -0.1 | -0.2 | -0.2 | -0.2 |
| **14 yzC** | -0.5 | -0.0 | -0.5 | -0.2 | -0.1 | -0.3 | -0.3 | -0.2 | -0.2 | -0.2 | -0.3 | -0.3 | -0.3 | -0.2 | -0.2 | -0.3 |
| **15 xzC** | 0.4 | 0.2 | 0.5 | 0.7 | 0.4 | 0.5 | 0.7 | 0.7 | 0.7 | 0.6 | 0.7 | 0.7 | 0.6 | 0.7 | 0.7 | 0.7 |
| **16 mot** | -0.3 | -0.3 | -0.4 | -0.8 | -0.5 | -0.5 | -0.8 | -0.9 | -0.8 | -0.6 | -0.8 | -0.8 | -0.7 | -0.8 | -0.8 | -0.7 |
| **17 xFA** | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.4 | 0.4 | 0.5 | 0.2 | 0.2 | 0.3 | 0.1 | 0.2 | 0.2 |
| **18 xFF** | 0.3 | 0.0 | 0.4 | 0.6 | 0.4 | 0.4 | 0.6 | 0.6 | 0.6 | 0.4 | 0.7 | 0.7 | 0.6 | 0.6 | 0.6 | 0.5 |
| **19 xS1** | 0.1 | 0.1 | 0.1 | 0.0 | 0.0 | 0.1 | 0.1 | 0.3 | 0.3 | 0.5 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.1 |
| **20 xS2** | 0.2 | 0.2 | 0.3 | 0.6 | 0.5 | 0.4 | 0.7 | 0.8 | 0.8 | 0.6 | 0.7 | 0.7 | 0.6 | 0.7 | 0.7 | 0.5 |
| **21 xS3** | 0.2 | 0.2 | 0.2 | 0.5 | 0.6 | 0.3 | 0.5 | 0.7 | 0.6 | 0.4 | 0.5 | 0.5 | 0.4 | 0.5 | 0.5 | 0.3 |
| **22 xS4** | 0.2 | 0.3 | 0.3 | 0.7 | 0.3 | 0.6 | 0.7 | 0.6 | 0.6 | 0.5 | 0.6 | 0.6 | 0.5 | 0.7 | 0.7 | 0.6 |
| **23 xE** | 0.2 | 0.3 | 0.2 | 0.5 | 0.4 | 0.4 | 0.5 | 0.8 | 0.8 | 0.8 | 0.5 | 0.5 | 0.6 | 0.5 | 0.5 | 0.5 |
| **24 yFA** | 0.6 | 0.0 | 0.7 | 0.5 | 0.4 | 0.4 | 0.5 | 0.6 | 0.5 | 0.4 | 0.7 | 0.7 | 0.7 | 0.6 | 0.5 | 0.5 |
| **25 yFF** | 0.0 | 0.4 | 0.1 | 0.5 | 0.3 | 0.3 | 0.5 | 0.5 | 0.4 | 0.3 | 0.4 | 0.4 | 0.4 | 0.5 | 0.5 | 0.4 |
| **26 yS1** | 0.6 | -0.0 | 0.7 | 0.5 | 0.4 | 0.4 | 0.5 | 0.5 | 0.5 | 0.4 | 0.7 | 0.7 | 0.7 | 0.6 | 0.5 | 0.5 |
| **27 yS2** | 0.3 | 0.1 | 0.4 | 0.6 | 0.4 | 0.4 | 0.6 | 0.6 | 0.6 | 0.4 | 0.8 | 0.8 | 0.7 | 0.6 | 0.6 | 0.5 |
| **28 yS3** | 0.3 | 0.1 | 0.3 | 0.4 | 0.5 | 0.3 | 0.4 | 0.5 | 0.5 | 0.3 | 0.7 | 0.6 | 0.5 | 0.5 | 0.4 | 0.3 |
| **29 yS4** | 0.3 | 0.2 | 0.3 | 0.5 | 0.4 | 0.4 | 0.5 | 0.5 | 0.5 | 0.4 | 0.6 | 0.6 | 0.6 | 0.5 | 0.5 | 0.5 |
| **30 yE** | 0.5 | 0.1 | 0.6 | 0.6 | 0.5 | 0.5 | 0.7 | 0.7 | 0.7 | 0.5 | 0.9 | 0.9 | 0.8 | 0.7 | 0.7 | 0.6 |
| **31 zFA** | 1.0 | -0.0 | 1.0 | 0.3 | 0.4 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| **32 zFF** | -0.0 | 1.0 | -0.0 | 0.4 | 0.3 | 0.3 | 0.4 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.4 | 0.4 | 0.3 |
| **33 zS1** | 1.0 | -0.0 | 1.0 | 0.5 | 0.4 | 0.4 | 0.5 | 0.4 | 0.4 | 0.3 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| **34 zS2** | 0.3 | 0.4 | 0.5 | 1.0 | 0.7 | 0.6 | 1.0 | 0.8 | 0.8 | 0.6 | 0.8 | 0.8 | 0.6 | 1.0 | 1.0 | 0.8 |
| **35 zS3** | 0.4 | 0.3 | 0.4 | 0.7 | 1.0 | 0.4 | 0.6 | 0.6 | 0.5 | 0.4 | 0.6 | 0.5 | 0.5 | 0.7 | 0.7 | 0.5 |
| **36 zS4** | 0.3 | 0.3 | 0.4 | 0.6 | 0.4 | 1.0 | 0.6 | 0.5 | 0.5 | 0.4 | 0.6 | 0.6 | 0.5 | 0.6 | 0.7 | 0.5 |
| **37 zE** | 0.3 | 0.4 | 0.5 | 1.0 | 0.6 | 0.6 | 1.0 | 0.8 | 0.8 | 0.6 | 0.8 | 0.8 | 0.7 | 1.0 | 1.0 | 0.8 |
| **38 xW1** | 0.3 | 0.3 | 0.4 | 0.8 | 0.6 | 0.5 | 0.8 | 1.0 | 1.0 | 0.7 | 0.8 | 0.8 | 0.7 | 0.9 | 0.8 | 0.7 |
| **39 xW2** | 0.3 | 0.3 | 0.4 | 0.8 | 0.5 | 0.5 | 0.8 | 1.0 | 1.0 | 0.7 | 0.8 | 0.8 | 0.7 | 0.8 | 0.8 | 0.7 |
| **40 xW3** | 0.2 | 0.2 | 0.3 | 0.6 | 0.4 | 0.4 | 0.6 | 0.7 | 0.7 | 1.0 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.5 |
| **41 yW1** | 0.4 | 0.2 | 0.5 | 0.8 | 0.6 | 0.6 | 0.8 | 0.8 | 0.8 | 0.6 | 1.0 | 1.0 | 0.8 | 0.8 | 0.8 | 0.7 |
| **42 yW2** | 0.4 | 0.2 | 0.5 | 0.8 | 0.5 | 0.6 | 0.8 | 0.8 | 0.8 | 0.6 | 1.0 | 1.0 | 0.8 | 0.8 | 0.8 | 0.7 |
| **43 yW3** | 0.4 | 0.2 | 0.5 | 0.6 | 0.5 | 0.5 | 0.7 | 0.7 | 0.7 | 0.6 | 0.8 | 0.8 | 1.0 | 0.7 | 0.7 | 0.7 |
| **44 zW1** | 0.4 | 0.4 | 0.5 | 1.0 | 0.7 | 0.6 | 1.0 | 0.9 | 0.8 | 0.6 | 0.8 | 0.8 | 0.7 | 1.0 | 1.0 | 0.8 |
| **45 zW2** | 0.4 | 0.4 | 0.5 | 1.0 | 0.7 | 0.7 | 1.0 | 0.8 | 0.8 | 0.6 | 0.8 | 0.8 | 0.7 | 1.0 | 1.0 | 0.8 |
| **46 zW3** | 0.4 | 0.3 | 0.5 | 0.8 | 0.5 | 0.5 | 0.8 | 0.7 | 0.7 | 0.5 | 0.7 | 0.7 | 0.7 | 0.8 | 0.8 | 1.0 |

# Appendix D

# Multiclass decision tree

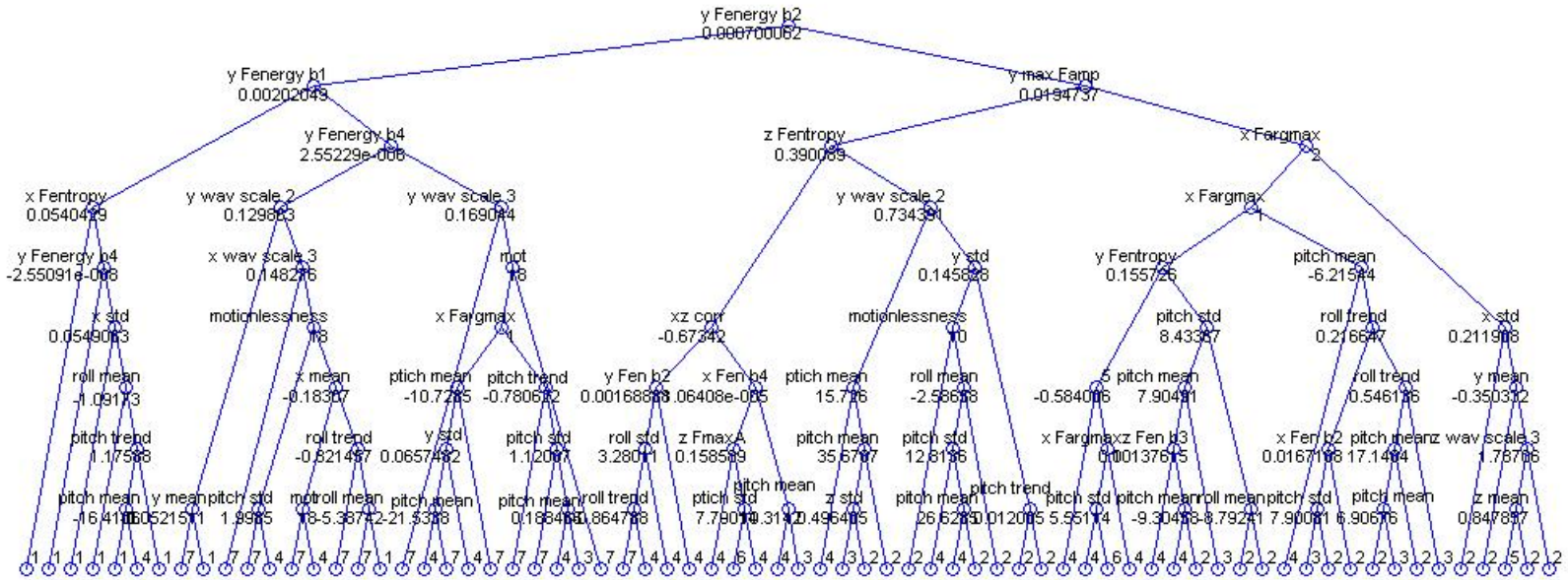



Figure D.1: Decision tree of the multiclass classifier. The numbers at each node represent the feature number and threshold. The endnodes show the decision class (1:stand, 2:flap, 3:soar, 4:walk, 5:fight,6:care,7:brood