UNIVERSITEIT VAN AMSTERDAM

Classifying Bird Behaviour
Based On Accelerometer Data

Moos HUETING 5967651

Robrecht JURRIAANS 5887380

Martijn v.d. VEEN 5964008

Bastiaan v.d. WEIJ 5922151

Abstract

In this paper we explore unsupervised learning methods using data gathered from small biologgers attached to the back of oystercatchers. We will extract statistical features from raw accelerometer data and supervised classification to evaluate our features. We will then apply an unsupervised learning algorithm to the data and evaluate the relations between the labeled and the clustered data.

February 2, 2011

Contents

1 Introduction

A relatively new way of observing bird behaviour is to equip birds with small tracking devices. These tracking devices can be used to capture the GPS location, speed, and tri-axial acceleration of the bird.

Apart from knowing where birds go, it would be interesting to know what kind of behaviour birds exhibit at any given moment. It has been shown that some behaviours can be deduced from accelerometer data [1]. The major problem with this is that it is unknown beforehand what kind of behaviour corresponds to what kind of accelerometer data. An expert might be able to "read" some behaviours off the data but if we want to automate classification we are left with an unsupervised learning task.

Roeland Bom [1] overcame this problem by observing birds equipped with tracking devices and labeling their behaviour. This approach was reasonably successful partly because the chosen birds, oystercatchers, will stay relatively nearby their nest location. For other bird species, observation may be harder. It would be interesting to see what information can be gathered with unsupervised methods from the biologgers data.

In this project we focused on gathering as much knowledge as possible from the accelerometer data of 40 of these oystercatchers. We split the project into a supervised part and an unsupervised part. In the next section we will discuss the approach we took to meet this goal.

2 Approach

The method used to describe the data sequences is feature based. First we will extract a set of features for each sequence of data and then we will use these features to differentiate between the sequences. We selected features based on the types of behaviour we would like to classify. Features describing relative stance and activity are likely to be useful to differentiate between certain behaviours. Furthermore, features describing the periodicity and repetition of the data are probably useful to classify behaviours such as flying.

Sequences of accelerometer data could be caused by different behaviours. Splitting the data of sequences in smaller sequences could become handy, especially if sequences become longer. Extracting features from smaller, overlapping parts of the original sequence could help in finding sudden changes in the signal. These points could be used to split the sequence. This is needed to allow for less variation within certain types of behaviour, and possibly finding more behaviours as well.

Once the features have been extracted we can train a classifier on the features to determine whether these features are sufficient to describe the different behaviours. For this the labeled dataset as given by Roeland Bom[1] is used. If the results are good enough, we can conclude that the features are descriptive enough to differentiate between the behaviours specified in [1]. [1]

Since the number of types of behaviours that can be found using only the accelerometer data is unknown, it is necessary to use a method that finds the optimal amount of clusters within the data. Once the clusters have been determined we can for each cluster find the sequence of data that is closest to its centroid. These sequences can then be used as prototypes for each type of behavior. It may very well be that certain clusters describe the same type of behavior differing on certain features only because of the configuration of the accelerometer.

[1] We suspect that the labels assigned by [1] do not necessarily correspond to clusters found by unsupervised algorithms.

Behaviour	**Frequency**
Forage	328
Handle	97
Body care	105
Stand	145
Sleep	63
Aggression	68
Fly	16
Walk	41
Sit	50

3 Method

3.1 Dataset

3.1.1 Flysafe database

The data used in this project is gathered from loggers attached to the back of a number of oystercatchers. The loggers weigh about 13.5 grams and can be attached to the back of the birds without them experiencing too much discomfort. The loggers carry a GPS sensor and an accelerometer sensor. The data is stored on internal storage until downloaded over Wifi. The GPS sensor is capable of logging the GPS-location every 3 seconds. The device is solar powered and therefore cannot continuously capture GPS and accelerometer data. The loggers log measurements at a maximum rate of every 3 seconds, but usually much slower depending on available power. Each measurement consists of a GPS location and 3 seconds of accelerometer data. During these 3 seconds that accelerometer's output is measured approximately 20 times per second. The loggers carry a GPS sensor and an accelerometer sensor. The data is stored on internal storage until downloaded over Wifi. The GPS sensor is capable of logging the GPS-location every three seconds. The device is solar powered and therefore cannot continuously capture GPS and accelerometer data. The loggers log measurements at a maximum rate of every three seconds, but usually much slower depending on available power. Each measurement consists of a GPS location and three seconds of accelerometer data. During these three seconds that accelerometer's output is measured approximately 20 times per second.

For this project we chose to use accelerometer data only. The data we have available is a series of measurements taken at a variable interval, consisting of a variable number of accelerometer readings:

$$\text{Dataset} = M_1, M_2, ..., M_N$$

$$M_i = \left(\begin{array}{c} \text{Date}_i, \text{Logger}_i, 1, A_x^1, A_y^1, A_z^1 \\ \text{Date}_i, \text{Logger}_i, 2, A_x^2, A_y^2, A_z^2 \\ ... \\ \text{Date}_i, \text{Logger}_i, n, A_x^n, A_y^n, A_z^n \end{array} \right)$$

Where M_i is one measurement and A_x^i is one x-acceleration reading, A_y^i an y-acceleration reading and A_z^i a z-acceleration reading.

We had access to the data from forty biologgers.

3.1.2 Observations by Roeland Bom

In our supervised learning experiments we used observations that Roeland Bom did in [1] on 3 oystercatchers with biologgers. These observations had timestamps that we used to find the corresponding measurements in the Flysafe database. The different behaviours and their number of observations are:

In [1] these classes were subdivided in more specific behaviours (such as foraging on land versus foraging in shallow water) but some of these behaviours had so few observations that we decided to use the more general classes as specified in figure 3.1.2.

3.2 Featureset

In this subsection we discuss the different features that we extracted from the raw accelerometer data. The accelerometer data consists of 3 discrete sequences denoting acceleration in the x, y and z directions. When a feature is extracted for each direction separately, i is used to denote the direction (i.e. μ_x denotes the average in the x direction). The features are obtained by examining one sequence of measurements.

Average

The averages can be used to determine the pose the bird is in. For instance, when the bird is standing still we expect the μ_z to be high and the μ_x to be negative.

$$\mu_i = \frac{1}{n}\sum_{j=1}^{n} i_j$$

Standard Deviation

The standard deviation tells us something about the amount and type of movement occurring. When the bird is flying we expect the σ_z to be fairly high in comparison to when the bird is standing still.

$$\sigma_i = \frac{1}{n}\sum_{j=1}^{n}(i_j - \mu_i)$$

Maximum & Range

The maximum value for each data sequence together with the range between the maximum and the minimum holds information on the spread of the data sequence.

Kurtosis

The standard deviation feature does not discriminate between a set of points with frequent small deviations and a set of points with few extreme deviations. Kurtosis fills this gap, giving a measure of the ‘peakedness’ of the data. For data with few extreme deviations, the kurtosis value is high, whereas the kurtosis value is low for data with relatively constant deviations.

$$D_i = \frac{\frac{1}{n}\sum_{j=1}^{n}(i - \mu_i)^4}{(\frac{1}{n}\sum_{j=1}^{n}(i - \mu_i)^2)^2}$$

Trend

The trend gives information about the relative progression of the movement and is calculated by taking the linear least square fit of the data and using only the linear coefficient. A positive trend indicates a rising value of the acceleration in the selected direction.

$$c = (A^T A)^{-1} A^T i$$

$$trend_i = c_1$$

Autocorrelation

When the bird is flying, we expect to see a repeating signal in the z direction. Autocorrelation gives a measure of how much the given signal repeats itself. Autocorrelation can be calculated for different intervals; we have calculated it for $n = 1, 2, 3, 4, 5$.

Average change

A sequence generated by a low-activity behaviour is expected to have a relatively constant signal. The average change is thus expected to be high for high-activity behaviour like flying and aggression, while it is expected to be low for activities like sitting and sleeping.

$$c_i = \frac{1}{N-1} \sum_{j=2}^{N} |i_{j-1} - i_j|$$

Fourier transform

The Fourier transform of a signal decomposes it into its constituent frequencies. For a clear repeating signal this means the maximum value of the Fourier transform conforms with the frequency of the input. This is another measure that we expect to be a good discriminator for flying behaviour.

Kinetic energy

Since the birds have not been weighed we can not determine the actual kinetic energy. However, it is possible to combine the 3 acceleration directions to give a representation of the kinetic energy.

$$\text{combinedacceleration} = \sqrt{x^2 + y^2 + z^2}$$

For this new sequence the average and the standard deviation are calculated and used as features.

Ratios

The ratio between 2 directions gives information about the stance and movement of the bird. Again the average and standard deviation are calculated for all 3 ratios.

$$\text{ratio}_{ij} = \frac{i}{j}$$

Interdirectional difference

The difference between the acceleration of two axes can tell something about the stance of the bird.

$$\text{sqdiff}_{ij} = (\text{avg}_i - \text{avg}_j)^2$$

3.3 Clustering

There are different methods available for clustering unlabeled data. Two well-known methods are K-means clustering and the Expectation Maximisation algorithm. As the implementation of EM in Weka has the option to find the optimal number of clusters using cross-validation. Therefore, we have chosen for this method over K-means clustering. Doing the same clustering with K-means and comparing the results could be an interesting experiment for the future.

Figure 1: Classification results for different datasets

Dataset	Featureset	Accuracy
calibrated-unfiltered	all	**67.03**
calibrated-unfiltered	best 10	64.73
calibrated-unfiltered	best 5	63.64
calibrated-filtered	all	66.03
calibrated-filtered	best 10	64.34
calibrated-filtered	best 5	64.68
calibrated-clipped-10	all	65.06
calibrated-clipped-20	all	**67.47**
calibrated-clipped-30	all	66.15
calibrated-clipped-40	all	67.03
calibrated-clipped-50	all	66.70

4 Experiments & Results

4.1 Preprocessing

The dataset contained a number of invalid values, which were removed. The accelerometer values are given in millivolts, they can be converted to g with the following calibration formula: $A_d = (M_d - O_d)/S_d$ where A_d is the acceleration in g in axis d (x, y or z), M_d is the measurement in millivolts in axis d, O_d is the the offset and S_d is the sensitivity. These values were not provided for every device. When absent, the factory defaults where used ($O = 0$ and $S = 1365$ for all axes).

After removing invalid values and calibration we are left with a set of measurements with an average of sixty acceleration triples (x, y, z acceleration specified in g) per measurement. However, some measurements had only a few acceleration triples and some had a lot more than sixty. To see if this influenced results we created a dataset where all measurements containing less than fifty or more than seventy acceleration triples were discarded. We will refer to this dataset as *calibrated-filtered* from now on. We will refer to the unfiltered dataset as *calibrated-unfiltered*.

To test how much the number of acceleration triples in one measurement influenced results we also created datasets with measurements cut off at ten, twenty, thirty, forty and fifty acceleration triples. This corresponds roughly to measurements of 0.5, 1, 1.5, 2 and 2.5 seconds since acceleration triples are recorded approximately twenty times per second. We will refer to these datasets as *calibrated-clipped-x* were x is the maximum number of acceleration triples per measurement in the dataset.

4.2 Supervised Classification

The first series of experiments involved testing how well our features performed on labeled measurements. We tested classification on our filtered, unfiltered and clipped datasets. We tried several learning algorithms including a J48 decisiontree learner, a multilayer perceptron, k-nearest neighbour (KNN) and naive Bayes classifier. Overall, KNN outperformed the other algorithms. While the J48 decisiontree learner performed marginally less than KNN, the naive Bayes classifier performed surprisingly poorly with accuracies roughly twenty percent lower than KNN.

Since KNN performed well we decided to use this algorithm exclusively for our experiments. We tested the best dataset with three different featuresets: all features, the best ten and the best five features ranked by their gain ratio. To validate the classifier we used 10-fold cross-validation. The results can be found in figure 1.

We achieved the best results using our calibrated-unfiltered dataset with all features. Reducing the features used to the best five interestingly lowered accuracy only by a few percent.

The class assignments made by KNN on our calibrated-unfiltered dataset using all features can be found in figure 2.

Figure 2: Class asignments for calibrated-unfiltered dataset using the KNN classifier

a	b	c	d	e	f	g	h	i	← classified as
257	45	7	9	0	6	2	1	1	a = Forage
69	25	1	1	0	1	0	0	0	b = Handle
13	2	71	7	4	3	0	4	1	c = Body care
16	1	5	104	0	9	1	7	2	d = Stand
0	0	5	1	55	0	0	0	2	e = Sleep
9	2	4	13	0	33	0	5	2	f = Aggression
2	0	0	3	0	1	9	1	0	g = Fly
8	0	5	12	1	2	1	11	1	h = Walk
1	0	1	4	1	0	0	1	42	i = Sit

Figure 3: Classification results with merged classes

Dataset	**Featureset**	**Accuracy**
walk-stand-merged	all	69.11
forage-handle-merged	all	80.61
forage-handle-and-walk-stand-merged	all	81.50

The results in 2 show that classes forage and handle as well as walk and sit are very hard to distinguish using our featureset. We tried creating a dataset containing only forage and handle or walk and stand behaviours but we were not able to achieve better results that way. We assumed that the difficulty classifying these behaviours was either a problem with our featureset or the two behaviours looked too similar from accelerometer data. When merging the classes our results obviously improved, see figure 3. Here we used the calibrated-unfiltered dataset and 10-fold cross-validation.

Since the best five features were for a large part responsible for the achieved accuracies it is interesting to see what these features are. We used the gain ratio algorithm with our calibrated-unfiltered dataset to find them, see figure 4(a).

Interestingly none of these features is related to the absolute or relative positions of the x, y and z acceleration plots. These absolute and relative positions tell us something about the orientation of the accelorometer which, apparently is not that important when classifying oystercatcher behaviour.

We suspected that the apparent unimportance of the absolute and relative plot positions was a result of the behaviours set we used. To test this we used the gain ratio algorithm on the calibrated-unfiltered dataset with all behaviours except walk, sleep, sit and stand removed. The new ordering can be found in 4(b).

The average x_avg feature now ranked best. This relates to body position in x axis. The second

Gain Ratio	Feature
0.4581	xyz_dev
0.4417	x_dev
0.4208	z_rng
0.4088	x_change
0.4027	z_dev
0.3837	z_max
0.3765	z_change
0.3762	y_dev
0.3663	x_rng
0.3274	x_max

(a) Best features for classification

Gain Ratio	Feature
0.5052	x_avg
0.4837	xzr_avg
0.4802	zxdiff_avg_sq
0.3766	x_max
0.3181	x_dev
0.2993	y_change
0.289	xyr_kur
0.2865	xyz_dev
0.2804	y_dev
0.276	x_change

(b) Best features for walk, sleep, sit and stand classification and third place are occupied by features realted to the position of the x and z plots relative to each other.

4.3 Clustering

The results rendered by the supervised classification methods described above tell us that the featureset we selected can be used quite effectively to classify the labels applied by Roeland. However, it might very well be possible to extract more or different clusters from the raw data.

In the unsupervised part of this project we used EM to find clusters in the unlabeled data. Running the algorithm on the labeled dataset renders a set of 17 distinct clusters. Each of these clusters is described by a set of mean values and standard deviation for each feature. These centroids are not (necessarily) existing data points. To get a more realistic idea of what kind of points belong to what cluster, we wrote a program that compares each point in a given dataset to each of the centroids rendered by EM and then labels it as belonging to the cluster to which the distance is the smallest. Then, for each cluster, we check which of these instances is the closest to its corresponding centroid. We labeled these instances as 'prototypes' for the corresponding cluster. In appendix A a graph for each cluster's prototype is shown.

As can be seen, the prototypes are, as expected, quite different. Furthermore, cluster 6 is quite clearly flying. When classifying the 'Fly' datapoints in the labeled set, almost every single one of these is classified into cluster 6. Furthermore it is apparent that Cluster 9 corresponds to low activity behaviour. However, we cannot fully relate each of the found clusters to one of Roelands defined behaviours. This was to be expected, because while the labels defined by Roeland may be interesting to biologists, there is no guarantee these behaviours are the main clusters found in the raw data of the accelerometer, which is simply a record of kinetic activity. This result can clearly be seen in figure 8 in the appendix. The clusters found by EM are plotted against the behaviours as defined by Roeland in the labeled dataset. We can clearly see that the only behaviour quite accurately matched by a cluster is 'Fly'. The low-activity behaviours ('Stand', 'Sit' and 'Sleep') are all quite well-represented in cluster 9. The rest of the behaviours are neatly spread across all clusters.

Clusters 7 and 12 are missing in the prototype section. This is because using the algorithm we implemented, not a single data point was assigned to these clusters. When comparing these clusters to the others, it turns out that cluster 7 is almost exactly the same as cluster 2, and cluster 12 is just as similar to cluster 5.

4.4 Splitting

A sequence of accelerometer data represents a discrete part of a continuous flow of different behaviours. Sequences could turn out to be taken during the same behaviour, but the sequence could represent transitions between different behaviours as well. The accelerometer devices allow for very long sequences, so a general method for splitting the sequences would be nice. Such a method could be used as a preprocessing stage and, as stated below, could be improved by using the weights of the features after clustering to split when changes occur in the features most important for classification.

The main idea of our splitting algorithm is that a sequence could be represented by our features well enough to be able to discriminate between the different clusters, and thus hopefully between the different behaviours. We take as an assumption that small sequences could be used to classify not significantly worse than longer sequences. Our clipping experiments show that this assumption is reasonable at least for sequences of half a second. The raw sequence then could be divided into smaller, overlapping sequences. For example, as show in figure 4.4, a sequence of three seconds could be divided in sequences of one second, starting at $0.0s$, $0.5s$, $1.0s$, $1.5s$ and $2.0s$. A bigger overlapping part will cause the calculated difference value to be smaller, while a smaller overlapping part causes sudden changes. The window size and shift size both influence the resolution of the possible split positions, which is discrete in this method. We calculate the features for each of these small windows.

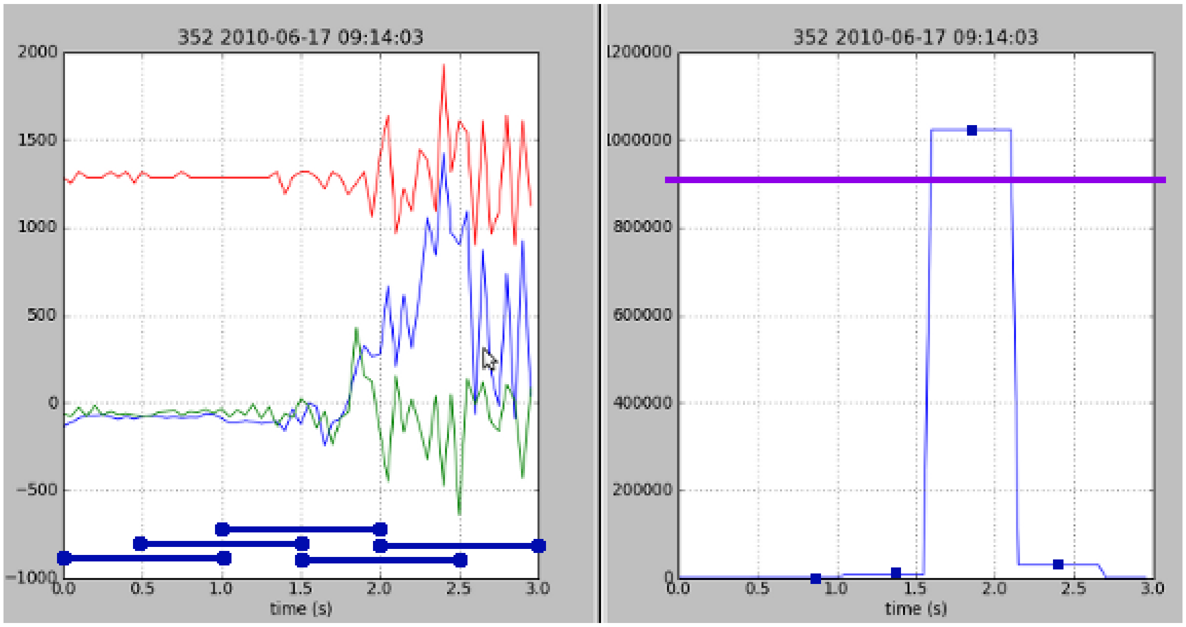


Figure 4: Splitting example

It is possible to classify each of those windows and check for transitions. However, we decided to preserve the splitting as a preprocessing step. This way, one could use standard toolkits like WEKA to classify without the need for calculations afterwards. Instead, we want a value representing the difference between each transition between two windows. One such method is based on the difference vector, which contains the difference between two windows for each feature. The length of the difference vector could be used as a measurement for the possibility of a changed cluster or behaviour. As an alternative, one could calculate the dot product between the feature vectors of the two windows. A threshold determines whether a split should be made or not. This threshold could be determined manually by inspecting the images with the calculated values, or automated using a supervised classification method. Experiments on the data using this simple splitting method show reasonable split points. Since not all features have the same range, one could divide each feature by its maximum to get a value between 0 and 1. For both methods the weight of the different features could be altered to reflect the weights of the most important features. After the process of splitting, the clustering could be done again with the splitted sequences.

Although the first experimental results were very promising and the need for splitting is certainly there, we did not go through the whole process of splitting the data and checking for differences in clustering and classifying. Further research on splitting is certainly possible, and could spawn interesting results.

5 Conclusions

Our selection of features has proved to be quite a good description of the raw data when it comes to classifying certain labels Roeland Bom[1] attached to the different points in his labeled set. We were able to differentiate 5 different behaviours with an accuracy of 81.50 percent (see section 4.2). However, the Forage and Handle as well as the Walk and Stand observations were too similar in the accelerometer data to classify. A possible explanation for why this is seems to be related to the fixed length of the measurements. The 3 seconds do not necessarily correspond to the length a certain behaviour is exhibited, nor does the start of the sequence correspond directly to the start of the behaviour. It seems that splitting sequences into subsequences might address this problem. We have seen that it is possible to use shorter samples of data without losing too much information about the characteristics of the behaviour. However, splitting is not enough. To be able to classify behaviours that correspond to sequences of lower level behaviours it is necessary to look at the patterns in the data.

During feature selection in our supervised experiments we have seen that the best features for classifying all behaviours are quite different from the best features for classifying a selected subset of behaviours. The x average was the best feature for differentiating between walk, stand, sit and sleep, while the xyz deviation was the best feature for classifying all behaviours. Since clustering on all features will weigh all features equally, the clusters that emerge did not correspond to behaviours. Perhaps a better approach to the unsupervised learning problem would be to first identify features that are relevant for classifying certain global groups of behaviours like high activity versus low activity behaviour and later split these groups into more specific behaviours. A good way to identify what features are useful for what behaviours would be to use labeled data if available.

The clusters that EM came up with are quite difficult to relate to the labels applied by Roeland [1]. More generally, it's hard to see what kind of behaviour is represented by the clusters. As a device recording activity and not behaviour, this was to be expected from the accelerometers. Apart from this difficulty, however, it is quite clear that the clusters are very different and thus represent different kinds of activity, even though they may correspond to the same kind of higher level activities.

6 Future Work

6.1 Interpret Labels

The clusters as found by the EM-algorithm are not yet interpreted. What needs to be done to interpret these clusters is find the closest match to the centroid of each cluster in the entire dataset. Once this has been done the extracted features along with the sequence of data can be interpreted by an expert. It may be necessary to include the extra available data such as the speed, location and the date as well as some meta data regarding the clusters, for instance the relative amount of sequences that belong to the cluster or the relative distances between clusters. Another possibility is to include the labeled data set[1] as this may help to understand what type of behavior is represented by the cluster. Using the labeled data it already became clear that cluster 6 as seen in figure 6 closely corresponds with the instances labeled as fly.

An important step in interpreting the labels might be to create new data that is combined with video footage so that, after the closest matches for each cluster have been determined for this new data, it is possible to actually view what the bird was doing at that exact time. This may be necessary since the accelerometer data sequences itself is just an abstract representation of the global body movements, and could turn out to be insufficient for an expert to determine the type of behaviours.

6.2 Experiment with clipped data

As we have seen, the clipped data is already quite descriptive of the behaviour of the bird. It could be interesting to see what kind of clusters EM or k-means would find with the clipped data and if these correspond to the clusters found in the unclipped data.

6.3 Detecting Sequences

By finding clusters in clipped data we throw away data. Splitting the data into smaller parts could be a better way of using the knowledge that small amounts of data are already quite descriptive. In the current approach the clusters are thought to represent types of behavior. However, it is possible to look at these clusters as being part of higher level behaviour, thus creating more clusters which could be combined as a sequence representing higher level behaviours. To do this it is necessary to look at the patterns in the data after splitting and clustering. Forage could consist of certain lower level behaviours such as walk, stand, pick and handle. It may be necessary to split the data sequences into subsequences. After splitting, the data can be analysed using for instance a Hidden Markov Model. An HMM seems to work very well for audio signals. By treating the accelerometer data in the same way as signals representing spoken audio, it could turn out to be possible to label the accelerometer data using similar techniques. More research is needed to determine the advantages of such methods.

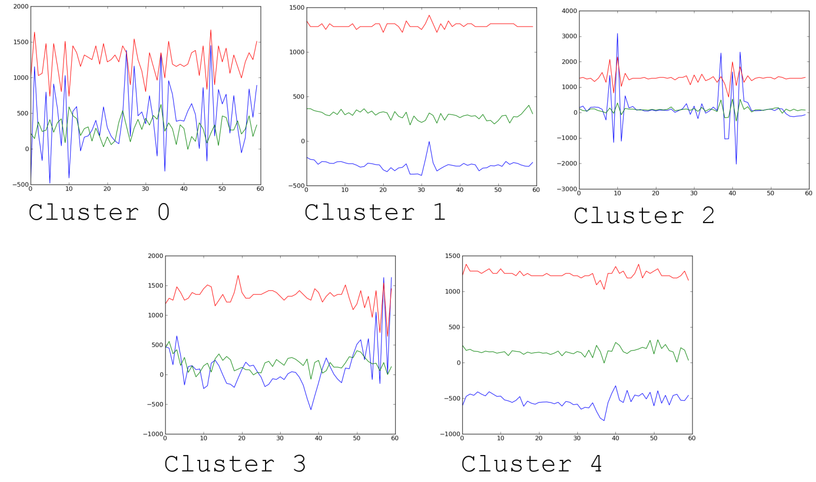
6.4 Other Birds

The method as it is now seems to be reasonably general. It could be used to determine behaviours in other birds and possibly other animals. It may be necessary to add new features that are better at describing certain patterns not found in the dataset of the oystercatchers. Perhaps it is even necessary to remove certain features that are irrelevant for the specific task, although these features will probably be ignored by the EM-algorithm. The clusters found for other animals may not correspond to the clusters found for the oystercatchers, since animals all have different bodies and types of movement and behaviour.

References

[1] Roeland A. Bom. Can speed and tri-axial acceleration measured by biologgers be used to classify oystercatcher behaviour? Master's thesis, University of Amsterdam.

A EM prototypes


Cluster 0

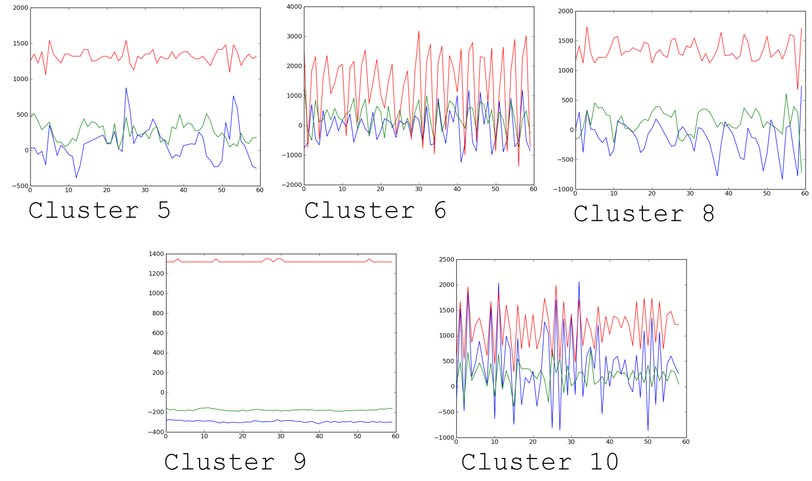
Cluster 1

Cluster 2

Cluster 3

Cluster 4

Figure 5: Cluster 0 to 4 found by EM


Cluster 5

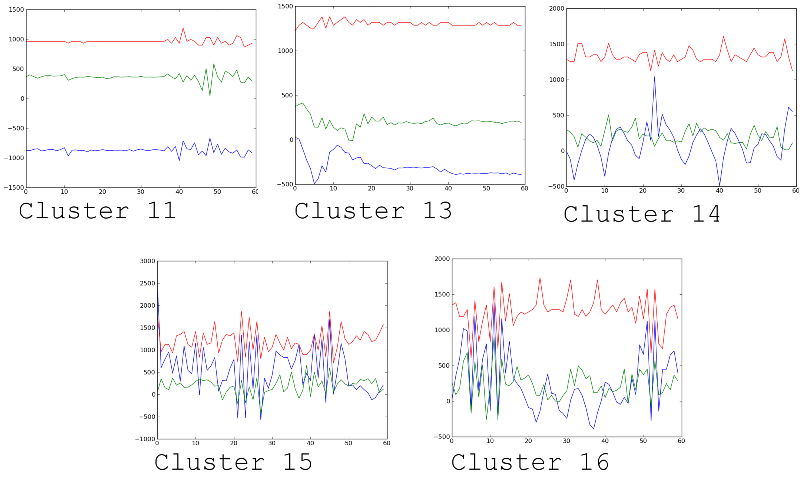
Cluster 6

Cluster 8

Cluster 9

Cluster 10

Figure 6: Cluster 5 to 10 found by EM, 7 missing



Cluster 11 Cluster 13 Cluster 14

Cluster 15 Cluster 16

Figure 7: Cluster 11 to 16 found by EM, 12 missing

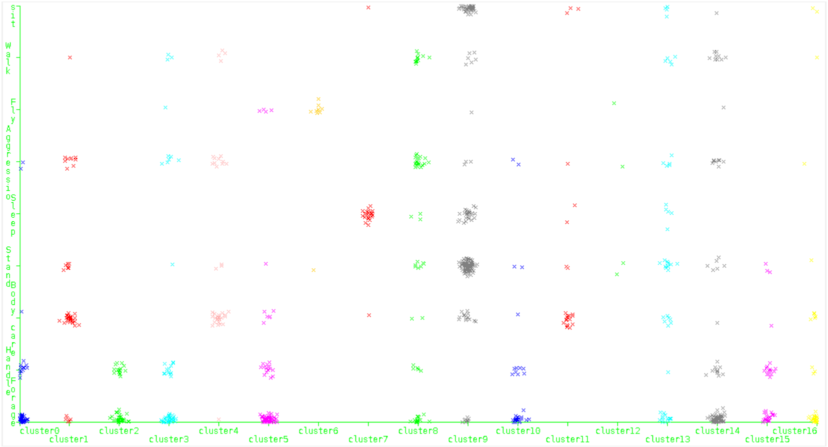


Figure 8: Assignments of behaviours to clusters