

# Biobots Motion Classification

Anamika Thakur  
North Carolina State University  
athakur3@ncsu.edu

Shankara Narayanan Sethuraman  
North Carolina State University  
ssethur2@ncsu.edu

Jordan Campbell  
North Carolina State University  
jhcampbe@ncsu.edu

## Abstract

*In this study, aggregated sensor data on cybernetic Madagascar hissing cockroaches, called 'biobots' is used to understand their motion. The motion of the Biobots can be classified into four states: stationary, random motion, clockwise motion and counter-clockwise motion. In this project, the classification process has been attempted with the aid of Hidden Markov Models.*

## 1. Introduction

The need to develop miniature, robotic entities that can carry a wide range of sensors and function effectively under complex, nonhomogeneous environments emerging after for example a natural disaster is very critical.

However, current technology falls short in offering mobile robotic agents that can do the job. Insects, on the other hand, exhibit an unmatched ability to navigate through a variety of environments via efficient locomotion.<sup>[2]</sup>

The Cyber Internet Networks project at the ARoS Laboratory, NCSU, uses cybernetic Madagascar hissing cockroaches, called 'biobots' and aggregates data from several biological and synthetic sensors periodically to analyze their natural movement.

The aim of this project is to attempt to understand the aggregated biobot data and develop an approach to classify their state.

In Project 2.1, two machine learning approaches were used for this classification task : Fine K-Nearest Neighbors and Cubic Support Vector Machine. They produced F1 scores of 0.7749 and 0.7962 respectively.

Since the data provided is a Time Series Data and the motion of biobots tend to mirror state transfers, we attempt in this project ( project 2.2 ) to model their motion through a Hidden Markov Model ( HMM ).

We expect ( intuitively ) that this procedure would produce significantly better results than the previous one.



Fig. 1 Cyborg Insect Networks<sup>[2]</sup>

## 2. Methodology

The aim of the project is to classify the different motions of the biobots. The input consists of around 10000 samples and each sample consisting of 42 features computer over 1 second window (with 75% overlap between measurements). The 42 features include accelerometer and gyroscope values of the biobots.

1:6 - Mean

7:12 - Variance

13:18 - Skewness

19:24 - Kurtosis

25:39 - Cross Correlation

40:42 - Gyro Energy

In addition, a ground truth table is also provided which consists of discrete values from 0-3 to indicate the state of the biobot at that particular timestamp.

The four states or the four motions that have been considered for classification are :

0 – Stationary / static

- 1 – Random motion in the arena
- 2 – moving clockwise on the boundary of the arena
- 3 – moving counter clockwise on the boundary of the arena

In the data provided, due to certain technical issues, some of the feature values were not received within the ranges acceptable to the sensors and hence were recorded as NaN ( i.e.) not-a-number. Of the 9999 input samples, only 9308 were observed to not have even a single occurrence of NaN in any of their features. That implies that ~ 7% of the available data would be ignored if we were to use only these data points.

As a solution, in the pre-processing step, each feature that was reported as NaN was replaced by the value of the same feature in the previous sample. The same is justified since each same has 75% overlap in time with both the sample preceding it and the one succeeding it. Therefore, such an approximation is not erroneous.

A hidden markov model (HMM) is one in which you observe a sequence of emissions, but do not know the sequence of states the model went through to generate the emissions. Analyses of hidden markov models seek to recover the sequence of states from the observed data.

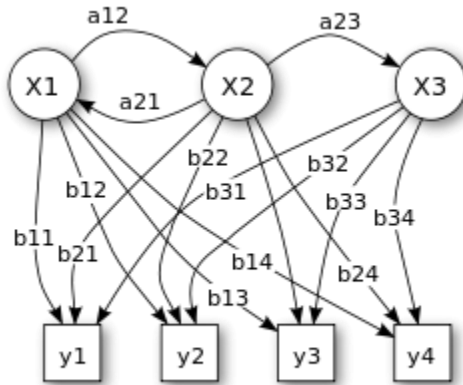


Fig 2. Hidden Markov Model (X – states; y – possible observation; a – state transition probabilities; b- output probabilities)<sup>[5]</sup>

The Hidden Markov Models Toolbox<sup>[6]</sup> from MATLAB is used to implement HMM on the dataset and classify the states of motion of the Biobots.

The function [ TRANS, ESTI ] = hmmestimate(seq,states) from the toolbox is used to estimate the emission and transmission probabilities for the dataset. The inputs seq and states are row vectors representing the input sequence and true states for the sequence.

In our dataset, we are provided with 42 features for a given time interval. However, the above function accepts only a single value per sample as input. Hence we are required to convert our Nx43 input data into a Nx1 array.

A good strategy for the same is to employ the classifiers from Project 2.1. We choose the Fine KNN classifier with Cosine Distance Metric and Squared-Inverse Distance Weight since it had the best performance.

The KNN algorithm is a type of instance based learning which classifies the test data points based on the closest training data points. It is the simplest of all machine learning algorithms. A KNN classifier uses K training samples that are closest to the test sample to classify it. Here the samples are classified based on their distance to points in a training dataset.<sup>[3]</sup>

The difference between different types of KNN algorithms is that they either vary in the number of neighbors used or in the distance metric or distance weight that is utilized.

The Fine, Medium and Coarse KNN algorithms all use Euclidean distance but have different number of neighbors.

Weighted KNN also uses Euclidean distance but uses squared inverses as distance weights. Cubic KNN uses Minkowski (cubic) distance and Cosine KNN uses cosine as the distance metric.

We are required to convert all N samples to a single value. For this we employ 5 fold cross-validation.

Cross validation is an evaluation model in which some part of the data is removed before training. This way only a part of the data set is used for training the learner. In a fivefold cross validation, the input data samples are randomly partitioned into five equal sized subsamples.

Out of the five subsamples, a single subsample is retained to be used for testing the model while four subsamples are used for training data. Then average error across all five trials is computed. The advantage of this method is that each data point is tested exactly once and gets to be trained four times. Due to this, the variance of the resulting estimate is reduced. In the fivefold cross validation, the subsamples are trained using K nearest neighbors algorithm.

The output obtained from the training using KNN is split into training ( 75% ) and testing ( 25% ) data.

The training data, transition and emission probabilities obtained from hmmestimate are used as an argument for the function hmmtrain. This function estimates the transition and emission probabilities for a Hidden Markov Model using the Baum Welch algorithm. It takes in three arguments – a sequence, and initial estimates of transition and emission probability matrices.

The transition and emission probabilities and the test data are used to calculate the posterior state probabilities using the MATLAB function `hmmdecode`. The output obtained is an array with the same length as that of the test data and one row for each state in the model. The  $(i,j)^{th}$  element of this matrix gives the probability that model is in state  $i$  at the  $j^{th}$  step.

We convert this  $M \times 4$  matrix into a single row vector for size  $M$  that contains the predicted state for every timestamp in the testing dataset. This is used to compute the confusion matrix, precision and recall and the F1 score.

### 3. Results

The confusion matrix is extracted using the posterior state probabilities and the ground truth labels for the testing data.

Confusion Matrix for HMM				
564	4	8	1	
43	1131	26	10	
5	15	428	0	
7	11	0	247	
F1 Score for HMM				
0.9461				

Fig 3 Confusion Matrix and F1 score for Hidden Markov Model

Confusion Matrix for Fine KNN				
1746	195	70	56	
282	4121	159	205	
89	216	1030	60	
84	271	38	1377	
F1 Score for Fine KNN				
0.8102				

Fig 4 Confusion Matrix and F1 score for Fine KNN

One can observe a significant improvement in the F1 score indicating that HMM is a better classifier for the purpose of classifying the motions of a biobot.

### 4. Analysis

As stated in the introduction, HMM was expected to perform significantly better than Fine KNN because the problem of classifying biobot motions was modeled using time series data the biobot movement can be represented as state transitions.

The problem with Fine KNN is not the algorithm itself, but rather the fact that it does not accurately model this particular application. KNN simply takes into consideration the euclidean distance between any given data point and the closest neighbor. Thus, if a point was moving clockwise based on the time series data, KNN will still classify the point as counterclockwise if the nearest

neighbor is in the classified as counterclockwise. This is shown pictorially in the figure below.

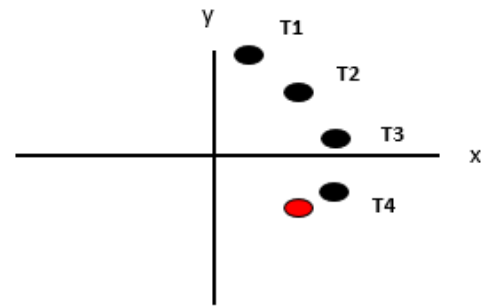


Figure 6 Problems with KNN

Image the x-y axis as a 2D plane on which the biobot moves. T1 is the first time point, T2 the second, T3 the third, and T4 the fourth. Using the time series data, it is easy to see this biobot is clearly moving clockwise. However, the red point has already been classified as counterclockwise. Thus, KNN makes no consideration of the time series data and falsely classifies this point as counterclockwise because the closest neighbor has been classified as counterclockwise.

No matter how many neighbors are picked, KNN is doomed to fail on certain points because it is the wrong model for this problem. HMM is the model we seek for this application. HMM works excellent for state machines where there is a time series data and transitions from one state to another are determined by probabilities. For example, consider the figure below. It does not truly represent the data used in our simulations, but is great for illustrative purposes.

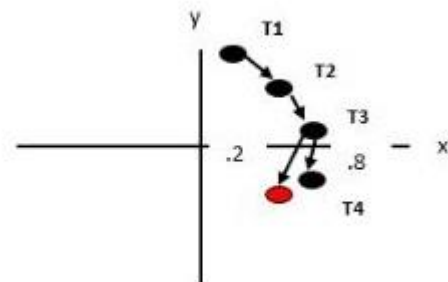


Figure 7 HMM representation of biobot movement

Now, the model has been changed to reflect a state machine where each point is state with a probability to

transition to the next state. Focus only on the transition from T3 to T4. This model delineates that after T3, there is a probability of 80% that the cockroach is classified as clockwise and a probability of 20% that the cockroach is classified as counterclockwise. HMM has successfully solved the issue KNN encountered on this same problem. Even though the counterclockwise point is closer neighbor to the point at T4, HMM uses the information from the previous time point to easily select the appropriate class.

## 5. Conclusion

In this project Hidden Markov Model was used to classify the motion of biobots into four different states. For this purpose, fivefold cross validation was used to train the input data using cosine KNN algorithm. The resulting output was divided into training and testing data and was used as input argument for the three different HMM functions used in matlab.

The F1 score obtained was used to evaluate the performance of HMM as a classifier and it should be noted that HMM was a significantly better classifier than Fine KNN and Cubic SVM used in project 2.1.

This improvement was expected not because HMM is necessarily a better classifier than KNN or SVM, but solely based on the fact that it is the correct model to use for biobot movement. Given a different classification task where we are not given time series data, it is impossible to model the problem as a state machine, and HMM would inevitably perform poorly on such a model. However, biobot movement can be accurately represented as a state machine given time series data because the only way to correctly classify the direction is relative to the previous position. KNN and SVM do not make use of the previous position in classification.

From the differences seen in project 2.1 it can be concluded that the accuracy of the classifier has been improved. This can be further enhanced by using a more accurate method than cubic KNN, including select features which are more useful, including the temporal structure of locomotion in the list of features, and making the number of samples per state more.

## References

- [1] "Choose Classifier Options." *Choose Classifier Options*. Mathworks, n.d. Web. 13 Nov. 2016. <<https://www.mathworks.com/help/stats/choose-a-classifier.html?requestedDomain=www.mathworks.com>>.
- [2] "Cyborg Insect Networks | ARoS Lab." ARoS Lab. N.p., n.d. Web. 13 Nov. 2016.
- [3] k-nearest neighbors algorithm ( [https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) )

[4] Definition of Hidden Markov Model (<http://jedlik.phy.bme.hu/~gerjanos/HMM/node4.html>)

[5] Hidden Markov Model ([https://en.wikipedia.org/wiki/Hidden\\_Markov\\_model](https://en.wikipedia.org/wiki/Hidden_Markov_model))

[6] <https://www.mathworks.com/help/stats/hidden-markov-models-hmm.html>