University of Sheffield

Data Analytics in Neuroscience

Aditya Chauhan

Supervisor: Eleni Vasilaki

A report submitted in fulfilment of the requirements
for the degree of MSc in Advanced Computer Science

in the

Department of Computer Science

September 12, 2018

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Aditya Chauhan

Signature: Aditya Chauhan

Date: September 12, 2018

Abstract

The Morris Water Maze is the most commonly used technique in behavioural neuroscience for the study of spatial learning in rodents. Over the years there have been many developments in methods to use the data collected from MWM for learning more about behavioural psychology of the rodents and other animals. These methods span from simplistic models of manual measurements to more sophisticated models based on machine learning techniques.

In this project we will device a more generalized and robust classification technique based on the *Hidden Markov Model(HMM)* which will show the most prominent behaviour patterns shown by rodents. Also we will compare the learning rate of two rodent groups, Control and Stress group.

Keywords: Morris Water Maze, Behavioural psychology, Motifs, HMM, Classification techniques, Trajectory segmentation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the year there have been a lot of interest in modelling of activities and architecture of the nervous system to behavioural outputs. These models have been developed by a significant attribution and complexity over time to get more precise and detailed understanding of behavioural outputs. The model developed and is being used since early 1980's to assess cognitive spatial memory is Morris Water Maze(MWM) described by Richard Morris[20] is the most widely used paradigm for spatial learning[26]. This model has developed significantly in estimation since it was introduced from manual observation to semi-automated[12][27] to automated classification[14]. The earlier MWM models are based on rodents(rats) but the advancement in technology has enabled the use of virtual forms of MWM than can be used directly on human subjects to comparatively assess human and rodent place navigation[25], compare spatial learning in sexes[2] and assessment of factors like stimuli and age on spatial navigation.It also helped with mapping of different areas of brain under these effects[8][10][24][15].

In typical MWM experiment, the rodent is placed in the circular pool filled with opaque water and sometimes even in the dark environment and it has to rely on it's senses to find the platform. After number of trial it is expected to find the platform with comparative ease based on its learning pattern[20]. This process of finding a platform is called escape latency and in order to understand this concept various classification methods are used to understand the trajectory of the rodent in order to find the platform. All these classifications are based on various factors like time spent by rodent in each quadrant of pool, the directionality of the movement and total swimming distance in each trial[5][16][20]. Some other factors involved are body temperature of rodent[17] and cumulative distance to the platform from the starting position of rodent which is calculated with a constant sampling rate[11][9]. Building on these parameters and to minimize human errors more profound semi-automated classification methods are developed[12][27]. In this method the rodents path is not classified as a whole but divided in to multiple segments and each segment is classified with a certain feature thus for a single path we get more precise classification model. In this model the segment overlap sometimes go upto 70% which increases the computation significantly. There is another model under development that works on the clustering algorithm with HMM and GMM classifiers[14]. This model works on clustering the segments of similar shape and structure under same classifier thus minimize the overlapping and enhancing the computation speed.

In this thesis we will try to develop the model based on the semi-automated(segment of paths) model and HMM-GMM classifiers. There will also be an implementation of some spatial-temporal factors(e.g speed and direction).

1.1 Aims and Objectives

The main aim of this project is to build a computation model for the understanding of the behavioural psychology of an animal when stressed by external factors. The model is based on the work of *Gehring et al*(2015)[12], *Vouros et al*(2017)[27], *Illouz et al*(2016)[14] and *Buchin et al*(2010)[7]. This model will classify the most prominent behaviour traits of rodent under stress.

In order to achieve this aim certain objectives need to be completed. The first objective is to go through the literature review and to deduce the modelling methods and techniques used for building the semi-automated and automated system and on how to integrate the spatial-temporal factors in it. This task will also familiarize with the tools used to develop and implement these models.

The second objective is to understand the mathematical interpretation of the models and creating a mathematical model based on the combination of semi-automated and automated models[12, 27, 14, 7]. This will also include the mathematical interpretation of clustering and Markov models.

The third objective will be to implement this model based on derived mathematical interpretation and classify the most attributed traits of rodent behaviour. This is the most important task of this project and will define the comparison basis with the existing models.

The fourth task comes in action if the model derived by the combination of semi-automated, automated and spatial-temporal factors doesn't gives a result classification that is more comprehensive than the earlier results. This will also serve as the baseline to set up if semi-automated model is the more accurate one and by what factor.

1.2 Overview of the Report

This report consists of six chapters including this as the first chapter. Chapter two focuses on the literature review which covers past research and models developed for behavioural psychology and gives a theoretical understanding of the developed models. This chapter also focuses on the shortcomings of the previous models and possible solutions to eliminate those shortcomings and develop a better understanding of this topic. Chapter three focuses on the methodology used to develop the model and than how that model is used to access the behavioral traits of rodents trajectories. This chapter also define the methods used to tweak a model in order to establish comparison between control and stress group of rodents. Finally there is a section defining the method used to calculate the efficiency of model. Chapter four focuses on the detailed analysis of results and related discussions. Chapter five concludes the project and describes the future significance of this model. Chapter six gives a list of project requirement and a disclaimer for the project.

Chapter 2

Literature Survey

2.1 Morris Water Maze

The study compiled in 1984 and published by R.Morris[20] defines a technique that is still used to develop models for the spatial learning and behavioural psychology. This method was devised in response to the controversies about the neural basis of spatial and working memory[21][22].

The essential features of this technique are: rodent is placed into a large circular pool of water and is a platform hidden with two different experimental scenarios, first by making it's top surface just beneath water and second by making water opaque and that serves as escape mechanism.

In MWM rodents start with the very random motions in different quadrants. Some circles around in the particular quadrant while some scans the sides of the maze. But in practise all these motions are very scattered but over time by spatial learning it minimizes the randomness in path and it takes less time to find the platform. Based on these observations there have been a development in specific characteristics of rats movement traits. In start these traits were manually observed and complete paths were classified in four different traits. These four basic observed trait features are *thigmotaxis , target scanning, incursion and Scanning*. As these are very discrete features compared to one another they are very easy to manually label. In *thigmotaxis* rats usually spends most of the time near the wall. In *target scanning* most of the time is spent on scanning area next to or surrounding the platform. In *incursion* most of the time is spent on moving inwards from the maze's wall. In *scanning* rat spends most of its time in a particular quadrant and don't reach towards the platform.

Due to the simplicity of this manual model defined by four behavioural traits and multiple experimental observations it is proved that these simple measures are not enough to quantify the wide range of other behavioural traits. In some cases it is observed that movement in the maze is the combination of multiple traits. In order to be able to better characterize the behaviour more sophisticated methods were proposed over the years. A scoring system of swimming paths was developed based on measures such as time rat spends in *right* or *wrong* quadrant[23]. Another model proposed shows that time spent next to platform correlates to the cumulative distance to the platform and not to escape latency[9]. A more accurate model was developed by dividing the swimming paths into seven different traits[13]. The traits of the path ranged from *thigmotaxis*(never finding platform) to *direct finding*(straight path to platform). Though this method improved the classification and gave a more profound correlation with escape latency, this model still worked on assigning one trait to one complete path thus rendering it insufficient as longer paths shows traits of more than one exploration strategy making this model more ambiguous in results. All these models so far are based on manual labelling of traits on swimming paths with single trait per path. The new *semi-automated* model was developed that increased number of traits to consider to eight and took in account the concept of multiple traits in single swimming path[12][27].

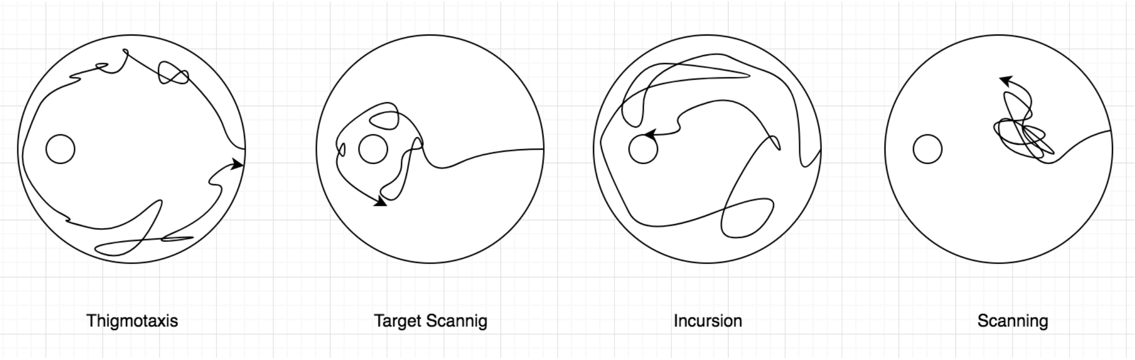


Figure 2.1: Swimming paths showing different types of behaviour traits

2.2 Semi-Automated model based on Morris Water Maze

This model is more granular classification method of swimming paths in MWM. To quantify changes in behaviours within a trial, the classification is done not to a complete path but at rather smaller segments of the complete swimming path. This method maps swimming paths to multiple traits thus making it possible to identify the subtle changes in behaviour between trials and among different animal groups.

There are eight different swimming path behaviour traits introduced in this method, four of which are the traits used in original MWM experiment[20]. Four other traits considered in this model are: ***Chaining Response***, concentric paths where animal memorizes the distance from the wall. ***Focused Search***, where animal limits it's search to very small area and sweeps them repeatedly. ***Self Orienting***, path where animal takes a full turn to orient itself. ***Scanning Surroundings***, where animal takes open paths passing through a critical regions around the platform.

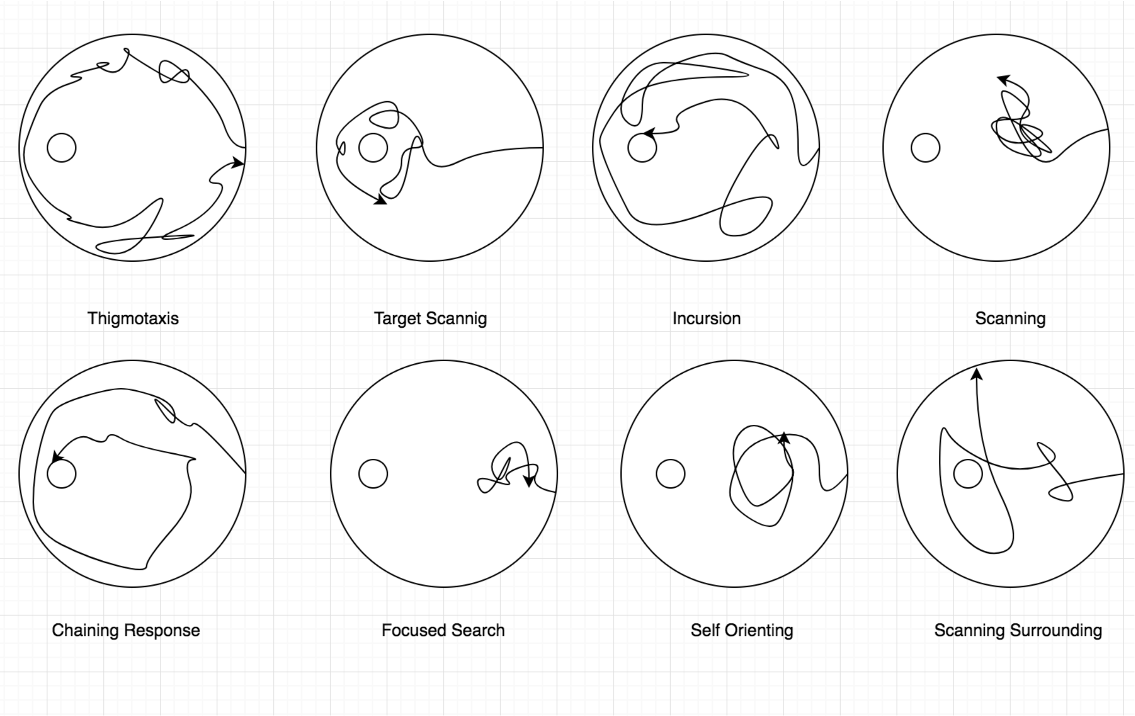


Figure 2.2: Swimming paths showing eight different traits used in development of semi-automated model

The segmentation of path in this method is achieved by custom analysis tools based on the clustering algorithm. These segments are same length and overlaps substantially thus making sure that classification is not affected by unfavourable segments. The segments overlap can range up to 70% making a dataset that needs to be classified quite large. Approximately 30,000 segments worth of data is considered in this method and as it is not possible to manually classify them so a semi-supervised model based on machine learning technique using clustering algorithm is implemented.

This method gives a quantitative behavioural difference in comparison with standard metrics of full swimming path. This method also gives a significant output to detect behavioural difference between two different animal groups. Most importantly it focuses on giving a discrete classification of traits applied by animals put under stress compared to animals not under stress(also called control group). This analysis also suggests that even though stressed animals are faster they still take longer to find the platform because of there use of low level strategies. This shows that they don't memorize the location of platform but memorize the distance of platform from walls and try to use that as a factor in finding a platform. These results are used in studies to prove that high level of stress lead to weak attention and frequent behavioural changes[1][18][19].

2.3 Automated model based on Morris Water Maze

This method works on more complex principals of machine learning where every step from data collection to data analysis to trait classification takes place using automated techniques. In this method, they developed a model to get a very precise estimate of the stereotyped behavioural components also known as *motifs*. Over years there have been some successful attempts at developing these full unsupervised completely automated and unbiased mathematical model to map out behavioural repositories in animals[3][6][28]. Though in these approaches the distinguishable and stereotyped motifs were identified properly which covered the superficial structure of motifs, they ignored the multi-scale embedding structure of behavioural motifs. Some of the earlier clustering algorithm approaches[12][27] have been quite successful but overlooked the transition dynamics between motifs.

In this new approach of the computational model was used that takes in account the existence, geometry and transition dynamics of the rat behavioural trajectories at *multiple timescales*. The model is based on *Hidden Markov Modelling* and segmentation decomposition procedure using *Principal Component Analysis (PCA)*. For comparison structure the *Gaussian Markov Model* and simple *Gaussian Model* is also implemented.

In theoretical approach a complete trajectory is decomposed to multiple small segments without overlap using PCA and each of these segments showing same structural composition on different timescales are than passed through HMM. Division of trajectory in segments is done based on the coordinates of each point forming the segment. Each timescale is the representation of direct relation with length and shape of segment. These sets of time scales when passed through the HMM produces a the sequence in which each segment occurs. Training this model with sufficient data produces a very accurate measure of the estimated sequence of the segments when applied to test data. Thus model due to no overlap in segments and due to its point to point segmentation gives a very accurate and computationally efficient result.

2.4 Summary

Over past few decades there have been an immense amount of research on the analysis of the behavioural features of humans and animals. Many techniques, from very simple human observations to very complicated computational models have been developed and are still under development. This review gives a very brief idea of some of the most important developments in this field over time and encapsulated the complexity of this research problem. To sum it up there is still a lot of development left to be done and lot of principals yet to be understood in this area of research.

Chapter 3

Methodology and Design

3.1 Introduction

This section defines the detailed understanding of the technical backbone of the project. It entails the significance of data used, on how that data is used and constrained into the developed model. It will also give a detailed understanding of the machine learning model developed in this project and its mathematical roots. Capping this chapter with the detailed understanding of the algorithm used and how that algorithm is tweaked to reach the best possible outcomes.

3.2 Data-sets Used for Modelling

This project serves as the next step in understanding the significance of the model developed by *Gehring et al*(2015)[12], *Vouros et al*(2017)[27], *Illouz et al*(2016)[14] and *Buchin et al*(2010)[7]. This model uses the clustering techniques with the weighted ranking method to segment the trajectory of the rodent based on the different behavioural patterns rodent shows when searching for the platform. The results generated by the above model is based on different criterion mainly acknowledging the *length of each segment* and *the overlap of each segment* with the previous segment. Both these features serves in developing a very significant model that can result in forming a complete original trajectory with minimum errors. The length of segments criterion used have three different values in the model to generate different sets of results. These lengths are 200cm, 250cm, 300cm with overlapping percentages of 70% and 90%.

The experiment was conducted on two different groups and there results are used by this classifier model. Group one is control group and group two is stress group. The control group is introduced to stress at peripubertal age and the results were to determine which group has more efficiency in finding the platform. Out of all the results obtained ones with the weighted ranking method were used which comprises of four data files(*one with 200cm segment with 70% overlap, two with 250cm segment and 70and 90% overlap and last one with 300cm segments with 70% overlap*). The data set contains set of attributes that are used in different combinations to get the results of different behavioural traits of rodent using the model developed in this project. The another dataset obtained and used is the time set which gives information about how much time each rodent spends in each state of the corresponding behaviour segments dataset.

Finally the results of this project's model are stored and can be used for the future development and analysis in rodents behaviours.

3.3 Algorithms and Techniques

The model developed is based on the Bayesian style models with a similarity to classification and clustering at a very core level. Though this model is based on HMM algorithm which is not classified primarily in any supervised or unsupervised learning.

HMM is a dynamic Bayesian model in which unlike other Markov models states are hidden and thus we also need an emission probabilities to train the model and produce best predictions.

For the implementation in our model the dataset mentioned in above section is used. The model uses the Viterbi style algorithm where going to next state from previous state is dependent on the emission probability of state in current step plus transition probability from the previous state. The algorithm is tweaked in a way to allow the most likely state sequence for the observation set. The dataset consists of sequence of 36 segments that represents the behaviours segments of rodent. The other information provided is *Animal ID, Trajectory ID, Trial No., Original Group and Target Group.* Each of these features are used in different combination to perform different analysis.

3.3.1 Primary model

In this model development a training of model is done on all four data-sets to produce a best state sequence that a rodent will take. In each data-set there are 54 rodents and each rodent go under 12 trails in the maze. Each trial of each rodent is considered a separate observation sequence to train the model. Once the model is trained the final path consisting of 36 behaviour traits is outputted.The algorithm calculates the transition probability, emission probability and average time at probabilistic scale for each segment all based on the observation sequence of segment and time datasets. These functions are than used by the Viterbi based algorithm in HMM to produce the best path.For best path we run algorithm for a complete length of observation sequence in accordance with 9 states each with the condition depending on emission probability for eat step in observation sequence. Conditions go forth as if emission probability is more than 0.8 than state with those emission probability is stored. If emission probability is between 0.1 to 0.8 than state with max emission probability plus transition probability from previous state is stored. If emission probability is very small(i.e below 0.1)than state with max emission plus transition plus average time is stored. The pseudo code of algorithm is defined below in Algorithm 1 and Algorithm 2:

3.3.2 Model to Compare best path between two target groups

The two animal groups are differentiated on the basis of the time at which they were originally subjected to stress. Target group 1 was subjected to stress at peripubertal age while target group 2 was subjected to the stress for the first time during MWM experiment. Each groups shows the definite difference in there ability to find and reach platform.

For each experiment on every test subject of either group, they were subjected to 12 trials. Though the rodents in group 2 do learn to find platform faster by trial 12 there results compared to the group 1 is less efficient. Not only do the group 1 rodents take less time to reach the platform they also memorize the direction and distance approximately to reach platform directly from the starting point and that two before they reach trial 12, sometimes even by trail 8 or 10. While group 1 rodents usually takes 11-12 trials to memorize the platform and in some cases even in 12 trails they don't reach the platform with extreme efficiency. To analyze the results of the difference between learning rate of the rodents of two target groups the above model is changed a little bit to produce two separate diagnosis.

Case One

In this case we take each of the four datasets file separately and combined and run the model to get best paths. This thing gives out 10 results of best paths 8(4+4) of which are the results for best paths of all group1 rodents(4 results for 4 datasets files) and best path for group2 similarly. Than combining the data of each file to make two separate datasets group1 and group2 and than finding best paths one for each. These results are than compared to analyze which group functions better under stress.

Case Two

In this case we take each rodent with different AnimalID find the best path based on all the 12 trials. This is again done on all four files separately and than combined for each

Algorithm 1 Best Path HMM

```
procedure TransitionProbability(obs)
    trans = numpy.array(9 X 9)
    sum_trans = numpy.array(9 X 1)
    trans_prob = numpy.array(9 X 9)
    for row in range(obs(all the rows)) do
        for col in range(obs(all the columns of each row)) do
            trans[obs[row][col]][obs[row][col + 1]]++
    for r in range(trans(all the rows)) do
        for c in range(trans(all the columns of each row)) do
            sum_trans[r]+ = trans[r][c]
    for r in range(trans(all the rows)) do
        for c in range(trans(all the columns of each row)) do
            trans_prob[r][c] = trans[r][c]/sum_trans[r]
    return trans_prob
procedure EmissionProbability(obs)
    ems = numpy.array(9 X 37)
    sum_ems = numpy.array(1 X 37)
    ems_prob = numpy.array(9 X 37)
    for row in range(obs(all the rows)) do
        for col in range(obs(all the columns of each row)) do
            ems[obs[row][col]][col]++
    for c in range(ems(all the columns)) do
        for r in range(ems(all the rows of each column)) do
            sum_ems[0][c]+ = ems[r][c]
    for r in range(ems(all the rows)) do
        for c in range(ems(all the columns of each row)) do
            ems_prob[r][c] = ems[r][c]/sum_ems[0][c]
    return ems_prob
procedure AverageTimeProbability(obs, obs_time)
    ems_time = numpy.array(9 X 37)
    sum_ems_time = numpy.array(1 X 37)
    ems_time_prob = numpy.array(9 X 37)
    for row in range(obs(all the rows)) do
        for col in range(obs(all the columns of each row)) do
            ems_time[obs[row][col]][col]+ = obs_time[r][c]
    for r in range(ems_time(all the rows)) do
        for c in range(ems_time(all the columns of each row)) do
            sum_ems_time[r]+ = ems_time[r][c]
    for r in range(ems_time(all the rows)) do
        for c in range(ems_time(all the columns of each row)) do
            ems_time_prob[r][c] = ems_time[r][c]/sum_ems_time[0][c]
    return ems_time_prob
```

Algorithm 2 Main function of Best Path HMM

```
procedure Main
    obs1 = (Matrix of observations from file 1)
    obs2 = (Matrix of observations from file 2)
    obs3 = (Matrix of observations from file 3)
    obs4 = (Matrix of observations from file 4)
    obs_time1 = (Matrix of observations from times file 1)
    obs_time2 = (Matrix of observations from times file 2)
    obs_time3 = (Matrix of observations from times file 3)
    obs_time4 = (Matrix of observations from times file 4)


    obs = combine all the obs1,2,3,4
    obs_time = combine all the obs_time1,2,3,4

    trans_mat = function TransitionProbability(obs)
    ems_mat = function EmissionProbability(obs)
    ems_mat_time = function AverageTimeProbability(obs, obs_time)
    path = numpy.array to save best state sequence

    for t in range of(length of observation sequence row) do
        for s in range of(number of states (i.e 9)) do
            if $ems\_mat > 0.8$ then
                $path[t] = numpy.argmax(ems\_mat)$
            if $0.8 > ems\_mat > 0.1$ then
                $path[t] = numpy.argmax(ems\_mat + trans\_mat)$
            if $ems\_mat < 0.1$ then
                $path[t] = numpy.argmax(ems\_mat + trans\_mat + ems\_mat\_time)$
    return path
```

rodent thus giving 270 best paths (54+54+54+54) each of 54 containing best paths of every rodent either belonging to group1 or group2. These results are compared for each file respectively and than last 54 results for combined data of all four files for each rodent and comparing between rodents of group1 and group2.

Both of the cases shows some definite differences in learning rate of rodents. Algorithms followed is similar as in Algorithm1 and Algorithm2 with just a difference in dataset used.

3.3.3 Model to analyze the learning rate of rodent through each trial

This model is to analyze the learning rate of each rodent rather than analyzing the learning rate of rodents in comparison to other. This way we get the estimation on how quickly they can adapt to stress though we can still make comparison but that point has been established in previous installment of the model.

In this development we take one trail at a time from each dataset file for each rodent thus giving us just a small dataset of observation(i.e 4 observations) and get the best path based on that. This way we will get 12 best state sequences one for each trial on every rodent and this gives us the efficiency graph of how fast rodent reached the platform.

This model is quiet simple to develop but gives a great insight on learning rate of rodents even in different target groups.

3.4 Efficiency and Error rate of the Model

This section describes the technique used to check the efficiency of the model and get the error percentages. The technique used to estimate this is ***K-Fold Cross Validation***. In this technique *K* by definition is basically in how many parts can the dataset be divided and than one set is chosen and model is trained on remaining *K-1* sets and than tested on the chosen set. This done done for K sets one at a time and the final error is averaged over the errors obtained in K iterations.

For this project the K-Fold cross validation is done for 4 different values of K each with the different significance. The basic algorithm for cross validation is describes in pseudo code Algorithm3 below. In this we basically compare the difference between the best paths of different groups and different K values. The four values of K for which model efficiency/error is tested is $K = 2,4,12,54$.

Algorithm 3 K-Fold Cross Validation

```
procedure CrossValidation(path1, path2)
    $count = 0$

    for $pos\ in\ range\ of(length\ of\ observation\ sequence\ row)$ do
        if $path1[pos] \neq path2[pos]$ then
            $count++$
    return count/37
```

3.4.1 2-Fold Cross Validation

In this case we use the result set obtained by the model describes in section 4.3.2. We have two datasets per file one for group1 and group2 and in case 1 and case 2 we calculate the best paths of combined group and individual AnimalID groups. Now using these two results in sync with each other we develop a 2-Fold cross validation. We take the 4 results for each file in case 1 for group1 and results from case2 for group 1 AnimalID rodents and find the error percentages. Similar for group2 and than we can do similar analysis for all the files combined as we have calculated the best path for combined file also in case 1 and 2. This gives an appropriate comparison basis between two groups.

3.4.2 4-Fold Cross Validation

This installment of the technique works on the primary model where we calculate the best path for each file separately and than all the files combined. Though for this case we have 4 data files making four separate datasets. Take one file and use its best path as a testing the model while other three files are used to train the model. This technique gives a good insight on segmentation techniques used for developing previous datasets used in this project model. This also serves as the proper technique to define the efficiency of this HMM based model developed in this project.

3.4.3 12-Fold Cross Validation

This installment of the technique is based on the model developed for section 4.3.3. I this case we have for each rodent a set of 4 observation sequences from four data files and each one results in a best path. Now we train this model for similar remaining 11 data set's best paths and test it against the remaining dataset best path we repeat it for all the 12 datasets and that too for all the rodent subjects. This results gives an efficiency on the trail level for each rodent.

3.4.4 54-Fold Cross Validation

This installment is the estimation of the efficiency of model compared to each rodent subject. First efficiency is calculated for each file separately and than combined for all the four data files. For each files there are 54 rodent subjects of which one is picked and model is trained on the remaining rodent subjects. The best path that obtained is evaluated against the best path obtained from the one rodent subject(it has 12 trials) that is being analyzed. This is done for all the subject and results are plotted. Once it is don separately for all the files now all the files are combined and similar analysis for a combination with 54 folds. This method gives a good insight on the efficiency of rodent compared to all the other rodents.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter gives a detailed sections for different results obtained by the model. Simultaneously there are sections with the results of different parameters used in the model and there detailed discussion on how these parameters are obtained. Finally this chapter also have a discussion on the significance of results obtained and their analysis on how these results shows the behavioral patterns of rodents.

4.2 Parameters for Model

4.2.1 Transition Probability

The transition probability for the model is calculated for estimating which state the rodent is going to be in the following the current state. The starting state is always **State 1** as rodent always starts with Thigmotaxis and next states follows. For calculating this probability we go through each observation sequence one by one and calculate the probability throughout the training set.

Formula to calculate the transition probability:

$$P(\frac{S_i}{S_j}) = \frac{\sum N(\frac{S_i}{S_j})}{\sum N(\frac{S_k}{S_j})} \tag{4.1}$$

where $N(\frac{S_i}{S_j})$ is the all the transitions from state S_j to S_i only, and $N(\frac{S_k}{S_j})$ is all the transitions from state S_j to all the states from 1-9. This gives the probability of transition from state S_j to S_i.

The results below shows the transition probabilities for individual dataset files and than all the files combined.

States	1	2	3	4	5	6	7	8	9
1	0.72	0.12	0.013	0.04	0.012	0.006	0.04	0.008	0.04
2	0.15	0.48	0.08	0.03	0.04	0.015	0.087	0.016	0.08
3	0.04	0.125	0.48	0.04	0.05	0.056	0.082	0.03	0.07
4	0.115	0.052	0.04	0.58	0.016	0.075	0.055	0.01	0.04
5	0.017	0.078	0.07	0.02	0.55	0.044	0.092	0.03	0.07
6	0.02	0.058	0.076	0.07	0.07	0.5	0.058	0.055	0.05
7	0.07	0.127	0.065	0.03	0.08	0.02	0.045	0.0558	0.06
8	0.02	0.03	0.039	0.031	0.03	0.05	0.05	0.58	0.14
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 4.1: Transition Probabilities based on dataset with segment length of 200cm with 90% overlap

States	1	2	3	4	5	6	7	8	9
1	0.68	0.13	0.0122	0.054	0.013	0.004	0.045	0.009	0.03
2	0.12	0.48	0.085	0.040	0.32	0.018	0.098	0.011	0.10
3	0.03	0.110	0.47	0.048	0.045	0.064	0.087	0.041	0.08
4	0.12	0.044	0.042	0.57	0.014	0.081	0.057	0.024	0.04
5	0.017	0.077	0.083	0.02	0.54	0.035	0.089	0.05	0.077
6	0.029	0.061	0.079	0.07	0.058	0.53	0.035	0.052	0.079
7	0.075	0.102	0.067	0.037	0.092	0.027	0.442	0.075	0.075
8	0.02	0.031	0.051	0.035	0.028	0.038	0.057	0.59	0.143
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 4.2: Transition Probabilities of group1 from data set of segments with 200cm length with 70% overlap

States	1	2	3	4	5	6	7	8	9
1	0.73	0.11	0.012	0.036	0.011	0.007	0.039	0.007	0.041
2	0.16	0.48	0.081	0.029	0.054	0.012	0.079	0.019	0.07
3	0.039	0.13	0.49	0.033	0.062	0.049	0.078	0.031	0.07
4	0.106	0.059	0.037	0.59	0.017	0.069	0.054	0.012	0.047
5	0.017	0.079	0.064	0.021	0.56	0.051	0.094	0.027	0.075
6	0.015	0.054	0.072	0.85	0.091	0.50	0.082	0.057	0.039
7	0.08	0.148	0.063	0.038	0.085	0.02	0.46	0.038	0.056
8	0.03	0.03	0.026	0.02	0.045	0.07	0.041	0.57	0.14
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 4.3: Transition Probabilities of group2 from data set of segments with 200cm length with 70% overlap

States	1	2	3	4	5	6	7	8	9
1	0.047	-0.026	0.0006	-0.017	-0.002	0.003	-0.005	-0.002	0.0032
2	0.042	-0.001	-0.003	-0.011	0.022	-0.005	-0.019	0.007	-0.03
3	0.0025	0.0271	0.0211	-0.0148	0.016	-0.015	-0.008	-0.009	-0.018
4	-0.017	0.015	-0.002	0.021	0.0025	-0.012	-0.002	-0.012	0.0075
5	-0.0006	0.002	-0.018	0.0006	0.021	0.015	0.005	-0.022	-0.022
6	-0.014	-0.006	-0.006	0.011	0.032	-0.029	0.046	0.004	-0.039
7	0.0058	0.045	-0.003	0.0002	-0.007	-0.0055	0.022	-0.037	-0.019
8	0.079	-0.0016	-0.024	-0.008	0.016	0.033	-0.015	-0.011	0.003
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 4.4: Transition Probabilities based on the difference of group1 and group2 transition probabilities.

Table 4.2 and 4.3 are based on the dataset for segments length of 200cm with 70% overlap. Table 4.3 is the difference of transition matrix group1 subtracted from group2. Now the results with positive value means the transition probability value of group2 was higher than one thus based on this group2 rodent in State 1(Thigmotaxis) stays in that state for longer duration compared to group1 rodent. While group1 rodent in state 1 transition to State2(Incursion), State4(Focused search) for most cases and even sometimes to Chaining response, Scanning surrounding or Target Scanning(States 5,7,8). Also from table 4.2 and 4.3 it is clear that rodent in state 1 from group1 always have higher probability of changing to other state compared to group2.

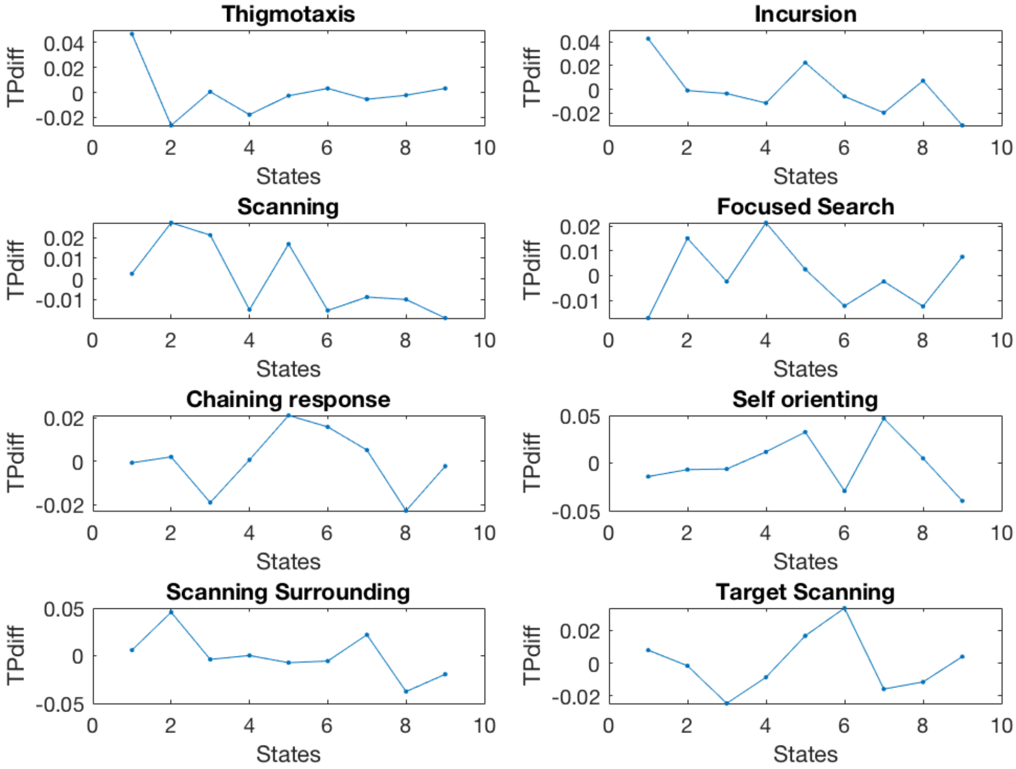


Figure 4.1: Graphs for each State representing the difference of transition probability between group1 and group2 (Trans_prob Group2 - Group1).

Based on this figure we can estimate which group transitions to which state at better rate. Every value positive in any subplot represents that group2 transition in that particular state from the state represented by subplots title. Thus subplot 1 shows that when in thigmotaxis group2 transitions into thigmotaxis for more segments thus spends longer duration in that. For Incursion it is observed that rodent in group2 going through incursion transits into Thigmotaxis more compared to group1.

While in target scanning group1 rodents go for very appropriate strategies like focused search around the platform as it was already in target scanning before that and the other strategy they rely on is scanning surrounding that too happen in platform quadrant which basically means that group1 rodents once near platform don't leave the platform quadrant while group1 two have relatively high probability to shift into thigmotaxis or self orienting and chaining response.

Similar extrapolation can be made for other significant behaviour traits like while in focused search group2 rodent tends to stay in focused search while group1 tends to shift to thigmotaxis to start moving towards platform or group1 shifts to self orienting in direction of platform. Similarly while in chaining response group1 do sometimes go to scanning but prominently to target scanning while group2 stays in chaining response or self orients itself.

While in scanning surrounding group2 rodents do sometimes starts doing incursion with approx. 5% more probability than group1 but in other retrospect once in scanning surroundings of platform both the groups usually stay in the quadrant though for group2 it usually have to self orient or do chaining response to reach scanning surrounding and to get to chaining response rodents in group2 usually goes through Incursion and scanning.

So by reading the above subplots properly we can estimate the proper transition sequence followed by a rodents in group1 and group2.

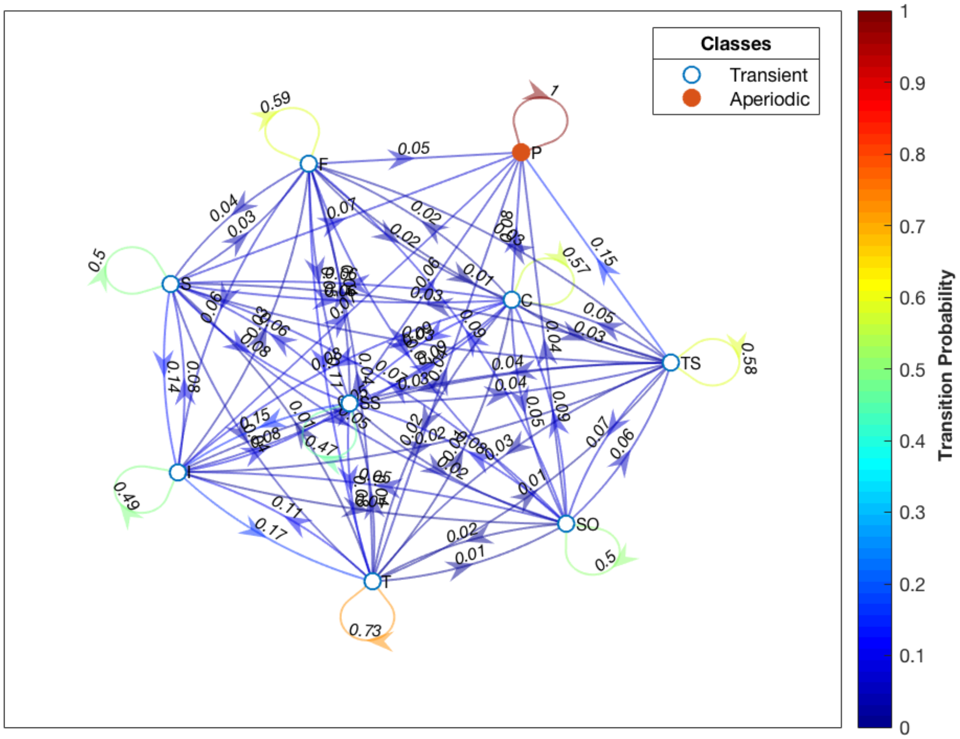


Figure 4.2: Markov chain directed graph plot for transition probabilities of group1.

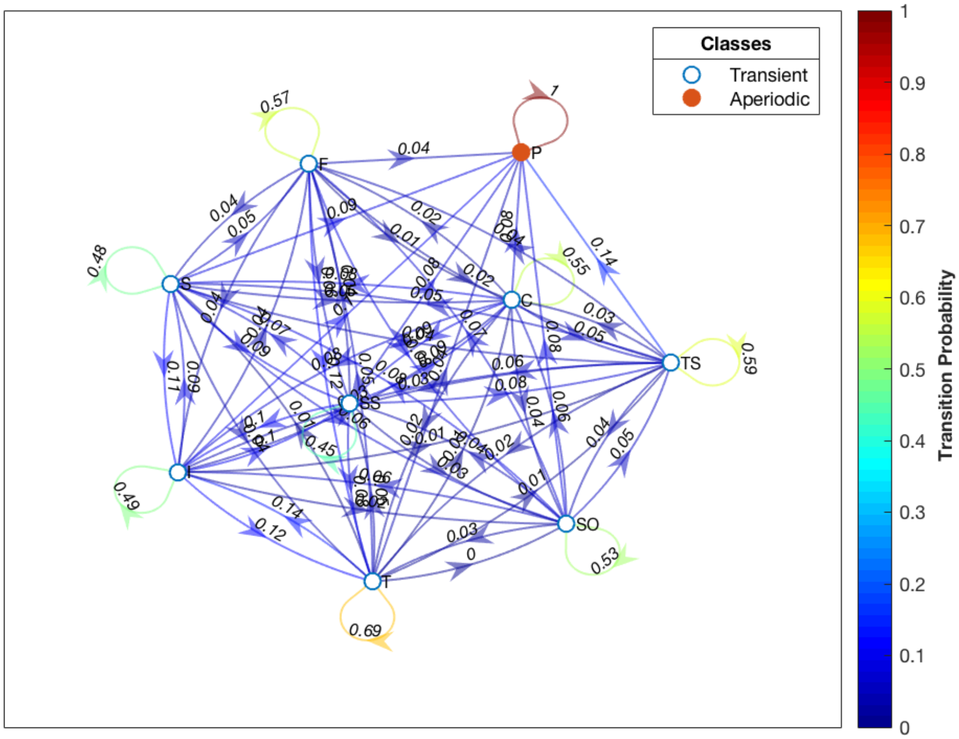


Figure 4.3: Markov chain directed graph plot for transition probabilities of group2.

4.2.2 Emission Probability

Emission probability is calculated by estimating which state the rodent is going to be in at a particular observation sequence. This basically the probability of emitting a particular hidden state in observation sequence. As rodent always start with Thigmotaxis than emission state at first observation sequence point is always **State 1**. For estimating the emission probability we go through complete observation sequence and calculate the probability of emitting particular state at each point.

$$P(O_i(S_j)) = \frac{\sum N(O_i(S_j))}{\sum N(O_i(S_k))} \tag{4.2}$$

where $N(O_i(S_j))$ is the number of times state S_j occurs at observation O_i and $N(O_i(S_k))$ is number of times each state S_k occurs at observation point O_i in observation sequence. This gives probability of getting state S_j at observation point O_i.

ObsSeq	1	2	3	4	5	6	7	8	9
1	0.09	0.08	0.1	0.11	0.12	0.13	0.12	0.12	0.12
2	0.12	0.1	0.09	0.11	0.1	0.1	0.1	0.09	0.1
3	0.18	0.16	0.1	0.09	0.1	0.08	0.06	0.06	0.06
4	0.07	0.08	0.05	0.04	0.05	0.06	0.06	0.07	0.06
5	0.07	0.08	0.1	0.1	0.11	0.08	0.06	0.06	0.06
6	0.13	0.14	0.1	0.08	0.06	0.06	0.05	0.04	0.03
7	0.15	0.13	0.1	0.1	0.08	0.07	0.07	0.07	0.06
8	0.04	0.07	0.12	0.09	0.06	0.04	0.05	0.04	0.04
9	0.15	0.15	0.22	0.29	0.32	0.37	0.42	0.44	0.47

ObsSeq	10	11	12	13	14	15	16	17	18
1	0.14	0.13	0.14	0.14	0.12	0.12	0.11	0.09	0.09
2	0.08	0.09	0.08	0.08	0.11	0.08	0.06	0.07	0.07
3	0.05	0.06	0.06	0.05	0.03	0.04	0.04	0.04	0.04
4	0.06	0.04	0.05	0.06	0.06	0.05	0.05	0.05	0.04
5	0.07	0.03	0.03	0.02	0.03	0.04	0.04	0.03	0.02
6	0.03	0.03	0.02	0.02	0.02	0.02	0.03	0.04	0.03
7	0.05	0.08	0.06	0.05	0.04	0.04	0.05	0.06	0.04
8	0.04	0.04	0.04	0.03	0.03	0.02	0.02	0.02	0.02
9	0.49	0.5	0.52	0.54	0.56	0.58	0.59	0.62	0.64

ObsSeq	19	20	21	22	23	24	25	26	27
1	0.08	0.08	0.09	0.08	0.07	0.06	0.06	0.06	0.04
2	0.07	0.06	0.06	0.06	0.05	0.04	0.04	0.02	0.02
3	0.03	0.03	0.03	0.02	0.01	0.02	0	0.01	0
4	0.05	0.05	0.04	0.04	0.03	0.02	0.01	0.01	0
5	0.02	0.03	0.03	0.02	0.02	0.02	0.01	0.01	0.01
6	0.04	0.02	0.01	0.01	0.01	0.01	0.01	0	0
7	0.04	0.04	0.03	0.02	0.02	0.03	0.02	0.02	0.01
8	0.01	0.01	0.01	0.01	0.01	0.01	0	0	0
9	0.67	0.69	0.71	0.74	0.77	0.81	0.84	0.87	0.9

ObsSeq	28	29	30	31	32	33	34	35	36
1	0.04	0.02	0.02	0.01	0.01	0	0	0	0
2	0.01	0.02	0.01	0.01	0.01	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0.01	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0.01	0.01	0.01	0.01	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0.93	0.95	0.96	0.97	0.98	0.99	0.99	0.99	1

Table 4.5: Emission Probabilities based on the dataset segments of length 200cm overlap 70%.

ObsSeq	1	2	3	4	5	6	7	8	9
1	0.1	0.09	0.1	0.12	0.13	0.14	0.13	0.13	0.13
2	0.15	0.12	0.1	0.11	0.12	0.12	0.12	0.11	0.11
3	0.14	0.14	0.12	0.1	0.09	0.08	0.07	0.06	0.05
4	0.07	0.08	0.07	0.04	0.04	0.05	0.06	0.06	0.05
5	0.04	0.06	0.08	0.08	0.07	0.05	0.04	0.04	0.04
6	0.12	0.12	0.1	0.08	0.07	0.06	0.05	0.04	0.03
7	0.15	0.15	0.13	0.11	0.1	0.09	0.08	0.08	0.07
8	0.04	0.06	0.08	0.07	0.05	0.04	0.04	0.04	0.04
9	0.18	0.18	0.22	0.29	0.32	0.37	0.42	0.44	0.47

ObsSeq	10	11	12	13	14	15	16	17	18
1	0.15	0.14	0.15	0.15	0.15	0.13	0.12	0.11	0.1
2	0.1	0.1	0.08	0.09	0.1	0.09	0.07	0.07	0.08
3	0.05	0.05	0.05	0.05	0.03	0.04	0.04	0.03	0.03
4	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.04	0.04
5	0.04	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02
6	0.03	0.03	0.02	0.02	0.02	0.02	0.03	0.03	0.03
7	0.07	0.08	0.07	0.06	0.06	0.06	0.07	0.07	0.06
8	0.03	0.03	0.03	0.03	0.02	0.01	0.01	0.01	0.01
9	0.49	0.5	0.52	0.54	0.56	0.58	0.59	0.62	0.64

ObsSeq	19	20	21	22	23	24	25	26	27
1	0.09	0.09	0.09	0.09	0.08	0.07	0.07	0.06	0.05
2	0.07	0.06	0.06	0.06	0.06	0.05	0.04	0.02	0.02
3	0.03	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0
4	0.04	0.04	0.03	0.03	0.02	0.02	0.01	0.01	0
5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0	0
6	0.03	0.02	0.01	0.01	0.01	0	0.01	0	0
7	0.05	0.05	0.05	0.04	0.03	0.03	0.03	0.02	0.02
8	0.01	0.01	0.01	0.01	0.01	0.01	0	0	0
9	0.67	0.69	0.71	0.74	0.77	0.81	0.84	0.87	0.9

ObsSeq	28	29	30	31	32	33	34	35	36
1	0.04	0.02	0.02	0.01	0.01	0	0	0	0
2	0.02	0.02	0.01	0.01	0.01	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0.01	0.01	0.01	0.01	0.01	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0.93	0.95	0.96	0.97	0.98	0.99	0.99	0.99	1

Table 4.6: Emission Probabilities based on all the datasets trained model.

4.2.3 Average Time

This parameter is significant when emission probability is very small(below 0.1) and at that stage it serves as a good deciding factor for choosing the correct state at that observation point in the sequence. Average time is calculated by using the time datasets files obtained from the previous model used for segmenting trajectories. For each point in observation sequence there is a corresponding time in the time dataset which tells how long rodent stayed in particular state at that observation point. We save that time as a time for a state in particular observation point thus making a **9 X 36** matrix. And as the probability goes below 0.1 it is used with transition probability and emission probability to give the best state at particular observation point.

Formula to calculate Average time :

$$T_A(S_i(O_j)) = \frac{\sum t_i(S_i(O_j))}{\sum t_k(S_k(O_j))} \tag{4.3}$$

where $t_i(S_i(O_j))$ represents sum of times corresponding to state S_i at observation point O_j in observation sequence and $t_k(S_k(O_j))$ represents sum of all the state times at observation O_j.

The result below shows the weighted Average time of based on dataset with 200cm segments length and 70% overlap.

ObsSeq	1	2	3	4	5	6	7	8	9
1	0.11	0.08	0.11	0.14	0.17	0.18	0.19	0.2	0.19
2	0.14	0.11	0.11	0.15	0.15	0.16	0.17	0.16	0.18
3	0.21	0.18	0.14	0.12	0.14	0.13	0.11	0.11	0.11
4	0.1	0.12	0.08	0.07	0.09	0.13	0.14	0.15	0.18
5	0.07	0.09	0.12	0.13	0.15	0.12	0.1	0.1	0.1
6	0.16	0.18	0.14	0.12	0.11	0.12	0.09	0.08	0.06
7	0.16	0.14	0.12	0.12	0.11	0.1	0.11	0.12	0.1
8	0.05	0.09	0.18	0.13	0.09	0.07	0.09	0.09	0.09
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ObsSeq	10	11	12	13	14	15	16	17	18
1	0.26	0.24	0.27	0.29	0.25	0.24	0.24	0.23	0.22
2	0.14	0.17	0.15	0.17	0.25	0.18	0.12	0.18	0.2
3	0.11	0.12	0.12	0.11	0.06	0.1	0.11	0.09	0.11
4	0.14	0.11	0.13	0.17	0.18	0.19	0.15	0.15	0.15
5	0.13	0.05	0.06	0.04	0.06	0.07	0.09	0.06	0.06
6	0.07	0.07	0.05	0.05	0.05	0.06	0.11	0.11	0.1
7	0.09	0.15	0.12	0.1	0.09	0.09	0.12	0.13	0.12
8	0.08	0.09	0.1	0.08	0.07	0.07	0.07	0.04	0.05
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ObsSeq	19	20	21	22	23	24	25	26	27
1	0.21	0.23	0.28	0.26	0.28	0.28	0.38	0.48	0.43
2	0.2	0.18	0.18	0.22	0.21	0.22	0.24	0.15	0.25
3	0.07	0.09	0.08	0.08	0.06	0.08	0.03	0.05	0.05
4	0.17	0.19	0.17	0.18	0.16	0.13	0.09	0.07	0.05
5	0.06	0.08	0.07	0.08	0.08	0.07	0.05	0.05	0.07
6	0.14	0.07	0.06	0.03	0.04	0.03	0.06	0.04	0.02
7	0.11	0.11	0.11	0.08	0.11	0.14	0.12	0.12	0.1
8	0.04	0.04	0.04	0.07	0.07	0.04	0.03	0.04	0.03
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ObsSeq	28	29	30	31	32	33	34	35	36
1	0.58	0.36	0.42	0.31	0.34	0.3	0.18	0.43	0.5
2	0.17	0.34	0.22	0.34	0.26	0.25	0.38	0.19	0.0
3	0.01	0.04	0.0	0.04	0.08	0.0	0.14	0.0	0.0
4	0.06	0.03	0.06	0.0	0.0	0.0	0.0	0.0	0.0
5	0.12	0.07	0.0	0.0	0.0	0.11	0.17	0.19	0.5
6	0.0	0.0	0.06	0.05	0.0	0.0	0.0	0.0	0.0
7	0.05	0.16	0.24	0.25	0.27	0.33	0.14	0.19	0.0
8	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4.7: Weighted Average time based on dataset with 200cm segments length and 70% overlap

4.3 Primary Model results and analysis

The primary model uses the data in above tables for **Transition probability, Emission probability and Average time** for training and than the trained model is used to predict the most probable state sequence. The model is trained by one file at a time which gives the most probable state sequence for each file and than combined to give the most probable state sequence for complete datasets.

4.3.1 Most Likely State Sequences

Result

Based on the above data most probable path based on:

File	PATH
1	[1 3 3 8 1 1 1 1 1 1 6 6 6 6 6 6 8 8 8 8 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
2	[1 3 7 7 1 1 1 1 1 1 6 5 5 6 5 8 8 8 8 8 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
3	[1 7 7 7 1 2 2 1 1 1 8 5 5 5 5 8 8 8 8 5 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
4	[1 7 7 7 1 1 1 1 1 1 5 5 5 5 5 8 8 8 8 8 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.8: Most probable state sequences for all the datasets files

Most probable path when the model is trained for all the dataset files is:

PATH
[1 3 7 7 1 1 1 1 1 1 8 5 5 5 5 8 8 8 8 8 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.9: Most probable state sequences by model trained for all the dataset files

Analysis

As all the datasets files used in this project are the variation of original trajectory just with different segment size and overlap percentage so the inference is gonna be very identical too.

All these most probable sequences starts with **State 1** which is bound to happen as in the initial training levels rodents always starts from near the wall thus **State 1(Thigmotaxis)** followed by the scanning of the regions or around the regions thus combination of **State 3 and 7** and than sometimes followed the incursion and more thigmotaxis(State 1 and 2) with combination of **State 5 and 6**(self orienting and chaining response) to move towards platform if it is going away from it and than finally **State 8**(target scanning) which is basically rodent has reached near the platform and scanning around platform itself. Following State 8 is usually **State -1**, this happens when rodent has reached the platform. Any States -1 after the first -1 is basically signifying that rodent is already on platform.

The most probable path when model is trained based on all the dataset files also exhibit same behaviour as the ones obtained for separate files as they all are the same trajectories with different segmentation behaviour.

4.3.2 Efficiency of Primary model

As the above section defines the most probable state sequences based on different dataset files and combination of all those files, there is still some differences in those sequences which is basically a defining factor for the efficiency check of the model. To text the model efficiency we validate it using *4-Fold Cross Validation* where we select three files for training the model and remaining one file for testing it. This way we get the efficiency corresponding to all the data files for this model.

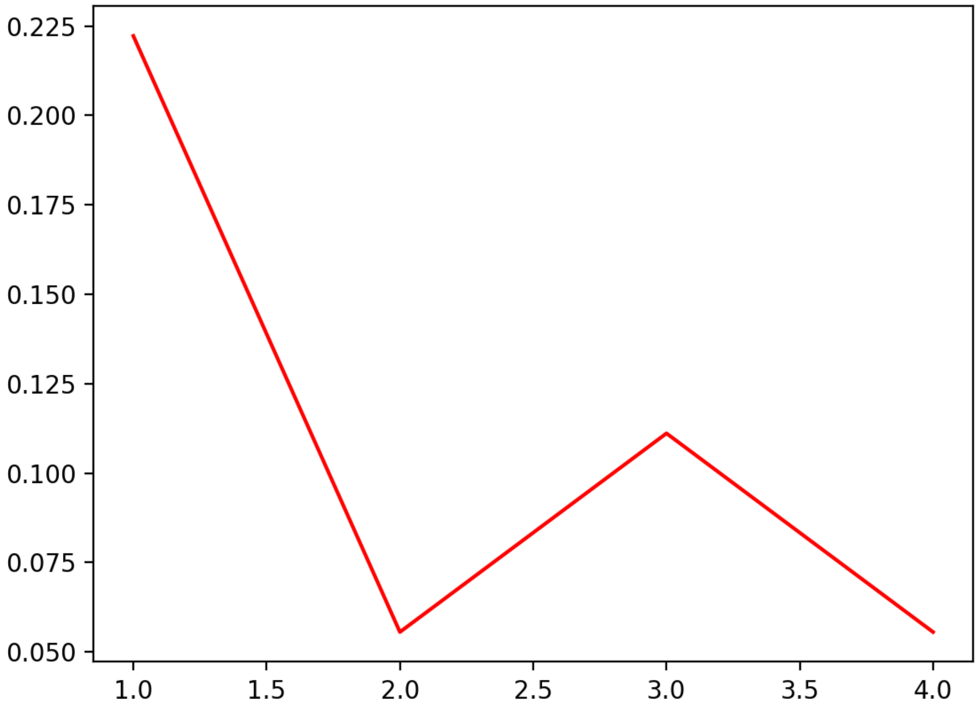


Figure 4.4: Graph of Error Percentages corresponding to each dataset file using 4-Fold Cross Validation

DatasetNo.(Segment_Len - Overlap%)	**ERROR%**
Dataset1(200cm-70%)	22.22%
Dataset2(250cm-70%)	5.56%
Dataset3(250cm-90%)	11.11%
Dataset4(300cm-70%)	5.56%

Table 4.10: Most probable state sequences by model trained for all the dataset files

Based on the above graph and table the average error percentage of the model is **11% approx.** and model works better when the segments are longer with less overlap as in these cases the datasets use for training have more efficient segments with more length and less overlap thus the most probable state sequence for these datasets also resembles closely to the most probable state sequence for the model trained by all four datasets.

4.4 Most probable path of two rodent groups and its analysis

For this analysis model was tweaked in a way such that each dataset file is divided in two sets one for control group(TargetID = 1) and second is stress group(TargetID = 2). The model than trains on these data sets and gives the most probable state sequence for each group similar result is obtained for all the data files combined. These results are in section below(Case One).

4.4.1 Case One

Result

GROUP	**PATH**
Group1	[1 6 6 8 7 1 2 2 6 6 6 5 6 5 8 6 7 5 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
Group2	[1 1 1 1 1 1 1 1 8 8 8 6 6 6 6 6 8 8 1 1 1 1 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.11: Most Probable state sequences for group1 and group2 from 200cm-70% file

GROUP	**PATH**
Group1	[1 6 7 7 2 1 1 2 5 5 4 5 5 6 5 8 5 5 7 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
Group2	[1 1 1 1 1 1 1 1 8 8 6 8 5 5 5 8 8 8 1 1 1 1 8 8 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.12: Most Probable state sequences for group1 and group2 from 250cm-70% file

GROUP	PATH
Group1	[1 7 7 7 7 2 2 2 6 5 4 5 6 5 5 8 8 5 8 8 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
Group2	[1 1 1 1 1 1 1 1 8 6 8 4 5 5 5 6 8 8 1 1 1 1 8 8 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.13: Most Probable state sequences for group1 and group2 from 250cm-90% file

GROUP	PATH
Group1	[1 7 7 7 1 1 1 1 5 5 5 5 5 5 5 8 8 8 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
Group2	[1 1 1 1 1 1 1 1 8 5 8 5 5 5 5 8 8 8 1 1 1 1 8 8 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.14: Most Probable state sequences for group1 and group2 from 300cm-70% file

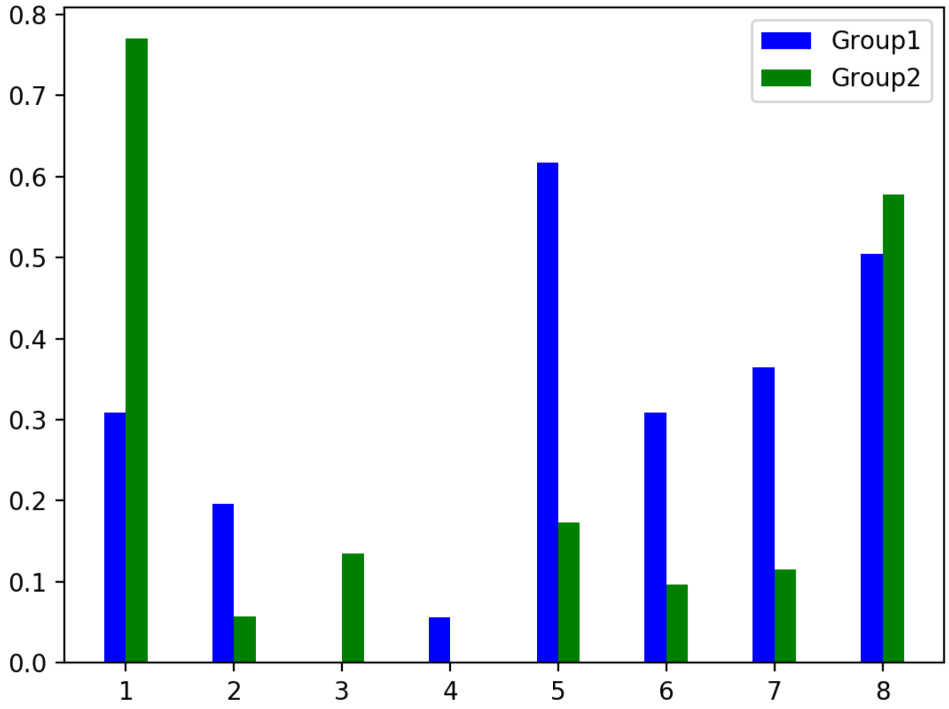


Figure 4.5: Plot showing different behaviour traits both the rodent group spends time in. In this plot X axis signifies 8 behaviour traits where **1 = Thigmotaxis, 2 = Incursion, 3 = Scanning, 4 = Focused search, 5 = Chaining response, 6= Self orienting, 7= Scanning surrounding, 8= Target scanning**.

Analysis

Based on Table 5.12,5.13,5.14,5.15 it is a clear observation that the rodents of **Group1(Control Group)** are more efficient at finding platform compared to **Group2**. In each case rodent in Group1 reaches the platform with a very systematic sequence of strategies and that too in less strategic steps compared to Group2. Thus there learning rate in the maze is very efficient compared to Group2.

The plot of the strategies(Fig 5.2) shows that Group 1 have much less transitions to State 1 compared to Group2 thus signifying that Group1 learns faster to move away from wall and towards the platform as in plot only group one does **Incursion(State 2)** which is why they move away from wall and towards platform. Also Group1 rodents performs more **self orienting** as rodents in group1 turns around if they are going in wrong direction. Finally compared to Group2, Group1 rodents do less **scanning of surroundings(State 7)** and more **Target Scanning(State 8)** because of their learning rate they estimate the location of platform and spend more time around it rather than in platform's whole quadrant. Thus this also gives a good insight on the learning rates of rodents in two separate groups.

4.4.2 Case Two

In this case the analysis is done for each rodent separately using 12 trials for each as training dataset.

Results

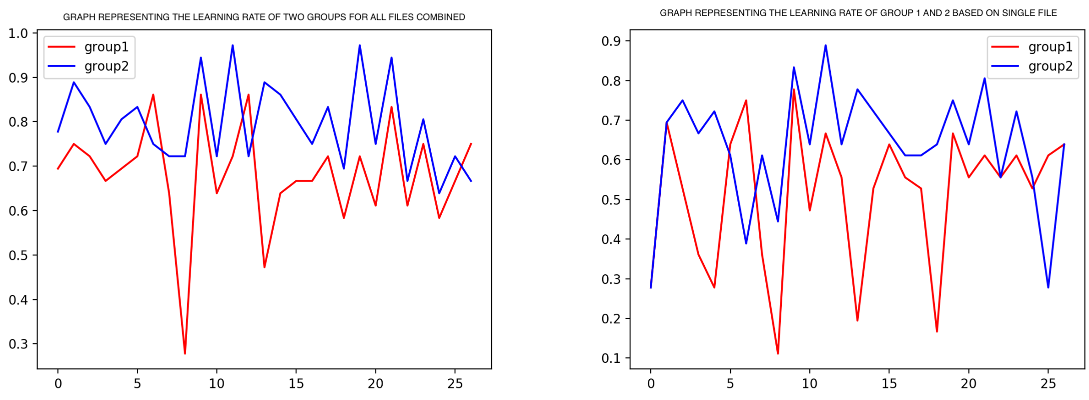


Figure 4.6: Plots representing the learning rate of each rodent from Group1 and Group2. The lower the value faster rodent reaches platform thus better learning rate. First plot is for all the data sets combined and second plot is for one data set file.

A.ID	State Sequence
87	[1 6 8 8 8 8 8 8 8 8 8 4 2 3 2 2 1 1 1 1 1 1 1 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
103	[1 2 8 8 7 7 6 6 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
91	[1 7 7 7 7 1 3 2 1 1 1 1 1 1 1 1 1 1 1 3 2 2 2 3 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
93	[1 8 8 8 8 4 3 3 1 1 1 2 2 2 3 3 2 2 5 5 5 5 5 5 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
95	[1 7 7 7 3 3 7 3 3 3 7 1 1 1 1 1 1 2 1 7 7 7 7 7 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
121	[1 2 2 1 4 4 4 4 1 1 1 1 1 1 1 1 1 2 7 7 7 7 7 8 2 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
99	[1 3 3 1 2 2 2 2 1 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 1 1 2 4 5 2 -1 -1 -1 -1 -1]
114	[1 6 6 6 6 2 3 3 3 3 2 2 3 3 4 1 1 1 1 1 1 1 1 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
101	[1 3 3 3 6 4 2 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
115	[1 2 7 2 2 2 7 7 7 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 2 2 1 1 1 3 1 7 -1 -1 -1 -1 -1]
43	[1 7 7 1 3 1 7 1 1 1 1 1 1 1 7 7 2 1 5 1 1 1 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
52	[1 1 3 3 3 2 2 2 2 1 7 1 1 2 1 1 1 2 4 1 7 2 2 2 1 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
49	[1 4 4 6 2 4 1 4 4 1 1 4 4 1 1 2 1 1 4 4 4 1 1 3 2 1 1 1 1 7 7 8 -1 -1 -1 -1 -1]
57	[1 3 3 8 8 6 1 1 5 1 1 3 3 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
59	[1 4 7 7 6 2 2 2 1 1 1 1 1 6 6 7 8 2 2 4 8 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
82	[1 6 2 8 8 8 1 1 1 7 1 1 1 1 1 1 2 2 2 2 1 1 1 2 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
65	[1 7 7 5 5 1 1 4 1 1 7 7 2 1 2 7 3 6 6 2 8 1 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
90	[1 3 4 4 3 1 1 7 7 7 8 3 3 3 3 6 6 7 1 7 2 2 2 2 2 7 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
67	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
50	[1 6 2 2 7 7 7 2 2 5 5 5 5 5 3 5 2 5 5 5 5 5 7 5 5 2 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
100	[1 1 1 2 2 1 1 7 7 7 4 4 4 1 1 1 1 4 4 4 4 1 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
75	[1 6 6 3 3 3 2 2 2 2 2 2 1 7 7 2 1 2 1 2 7 1 1 2 1 1 1 1 7 7 7 -1 -1 -1 -1 -1 -1]
83	[1 2 2 7 7 7 2 2 2 8 3 3 4 4 1 1 7 7 2 2 6 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
71	[1 7 7 7 1 4 4 4 1 1 1 1 1 1 1 1 1 7 7 1 1 1 1 2 3 3 3 3 -1 -1 -1 -1 -1 -1 -1 -1 -1]
111	[1 3 3 1 1 3 3 3 7 2 2 2 2 1 1 1 1 1 1 3 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
61	[1 3 3 5 2 1 1 2 2 2 7 7 4 4 7 7 7 3 3 1 4 4 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
94	[1 1 1 1 5 5 2 2 8 4 2 1 1 1 1 1 7 3 2 2 2 2 2 3 5 8 8 7 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.15: Most Probable state sequences for **Group1** per rodent for all data files combined.

A.ID	State Sequence
88	[1 7 7 7 1 7 7 3 1 1 2 4 4 3 3 2 2 2 2 3 1 1 1 1 3 1 1 2 -1 -1 -1 -1 -1 -1 -1 -1]
104	[1 3 8 3 2 2 1 1 1 1 1 7 1 1 1 7 7 1 2 1 1 1 2 2 2 1 1 1 1 1 1 1 1 -1 -1 -1 -1]
90	[1 3 3 7 7 7 2 2 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 2 2 -1 -1 -1 -1 -1]
106	[1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1]
108	[1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 7 2 2 2 1 1 7 1 1 1 -1 -1 -1 -1 -1 -1 -1]
118	[1 3 2 2 2 2 1 1 4 3 3 3 3 7 1 1 1 1 1 1 1 1 4 5 5 2 1 1 1 1 1 -1 -1 -1 -1 -1]
98	[1 7 7 7 5 5 5 1 1 6 6 1 1 1 2 4 4 4 4 4 4 4 4 4 4 4 4 4 -1 -1 -1 -1 -1 -1 -1 -1]
113	[1 7 7 7 1 4 2 2 2 1 1 1 1 2 2 1 2 2 7 4 1 2 2 3 4 5 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
100	[1 3 3 3 3 3 2 2 2 2 2 7 7 2 2 7 7 1 3 3 3 3 3 3 3 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
44	[1 3 3 5 5 5 5 7 2 7 7 7 2 2 2 5 5 5 2 7 7 2 1 1 7 7 1 2 1 1 1 3 3 3 1 -1 -1]
53	[1 6 6 6 6 2 2 2 2 2 2 2 2 2 4 4 7 7 7 7 7 7 7 7 3 3 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
50	[1 6 1 1 1 1 5 5 4 4 7 7 7 4 2 1 1 1 1 1 1 1 2 1 1 1 1 2 7 1 1 1 7 4 4 3 -1]
58	[1 2 2 2 2 2 4 4 7 1 1 1 1 8 8 4 4 4 1 1 1 2 1 7 3 1 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
60	[1 7 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1]
83	[1 5 5 2 2 3 2 2 2 2 2 7 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 3 1 1 1 -1 -1 -1 -1 -1]
67	[1 6 6 4 4 4 1 1 1 1 1 1 1 1 1 1 4 4 1 1 1 1 8 1 1 2 2 1 1 1 -1 -1 -1 -1 -1 -1 -1]
76	[1 7 7 1 1 8 8 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 7 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
71	[1 6 3 3 2 2 2 2 2 2 1 1 1 2 2 1 1 1 1 1 7 4 1 3 2 2 3 7 2 3 6 -1 -1 -1 -1 -1]
52	[1 2 7 5 5 5 7 7 5 2 2 2 7 2 2 2 1 7 7 7 2 2 4 4 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
92	[1 2 3 3 7 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 7 4 3 1 1 1 1 1 1 -1]
69	[1 7 7 7 2 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 1 1 1 1 2 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
57	[1 3 3 3 3 3 3 3 3 7 2 2 2 2 2 7 7 7 7 2 2 1 3 3 7 7 7 5 1 7 3 1 7 7 7 -1 -1]
107	[1 3 6 6 6 4 2 2 2 2 3 7 1 3 3 3 7 7 7 2 2 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
73	[1 2 7 7 7 1 1 1 7 1 2 7 7 7 5 5 5 7 7 1 7 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1]
81	[1 3 3 3 3 1 1 5 7 7 7 7 7 2 2 3 2 2 2 1 1 3 6 3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
63	[1 1 1 1 1 1 1 1 1 1 2 3 3 2 2 2 2 2 2 3 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
96	[1 7 3 1 6 6 6 6 6 2 2 6 4 4 4 4 4 1 1 7 1 2 2 2 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.16: Most Probable state sequences for **Group2** per rodent for all data files combined.

Analysis

Based on the plots and tables above there is a definite observation that every rodent individually too, from Group1 performs better in finding the platform compared to Group2 rodent. In both the plots red line represents Group1 and in both cases red line is mostly below blue line which mean that Group1 rodent reached the platform at these steps of observation sequence and on the other hand blue line in both cases represents Group2 where almost each rodent reaches after in later steps of observation sequence and also both the line are separated by quite a difference thus efficiency of Group1 in finding platform is much better than Group2.

4.5 Most probable path for each trial

For this analysis model was tweaked in a way to consider all four dataset files simultaneously. The training data for each case is very small as we take one observation sequence for each rodent and that too for each trial i.e we get the total of four observation sequence to train the model and get most probable state sequence for particular rodent at particular trial. This analysis gives a good view of how through each trial rodent is learning to find platform faster.

A-T	PATH
87-1	[1 2 1 7 7 7 1 4 4 1 1 1 1 4 4 4 7 6 4 4 1 1 1 1 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-2	[1 8 8 8 3 3 1 8 8 8 8 8 1 1 1 1 4 4 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-3	[1 3 2 2 7 8 8 8 8 8 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-4	[1 6 6 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-5	[1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-6	[1 7 8 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-7	[1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-8	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-9	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-10	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-11	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
87-12	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.17: Most Probable sequence for AnimalID 87 of Group1 for all trials

A-T	PATH
104-1	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-2	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1]
104-3	[1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-4	[1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-5	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-6	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-7	[1 3 3 3 3 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-8	[1 6 6 7 7 2 2 3 5 2 7 7 2 2 2 7 7 1 2 2 1 1 2 2 2 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-9	[1 8 8 1 3 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-10	[1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-11	[1 8 8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
104-12	[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]

Table 4.18: Most Probable sequence for AnimalID 104 of Group2 for all trials

These tabular results shows a very clear comparison in the ability of the rodent of two different groups to find platform. The group1 rodent(AnimalID 87) which is introduced to stress at peripubertal age, performs with some understanding and proper strategy to find platform that too from trail 1 and through observation it can be observed that by trial 6-7 all the rodent of group 1 gets a general idea of the platform's position in the maze. By trial 8-10 the rodent in group1 travel straight to platforms in most cases.

Taking the same account for group2 in most case each rodent in this group start with thigmotaxis and till almost trial 6 they travel around the wall than slowly learning a little bit about mazes and make their way towards the platform. Sometimes even after trial 6 rodent's motion is a little random. Considering all these situations it takes a rodent of group2 around 11-12 trials to reach the platform.

Both these tables(5.18,5.19) is just one AnimalID rodent from each group. The complete results are available in Appendix.

4.5.1 Strategies per trial

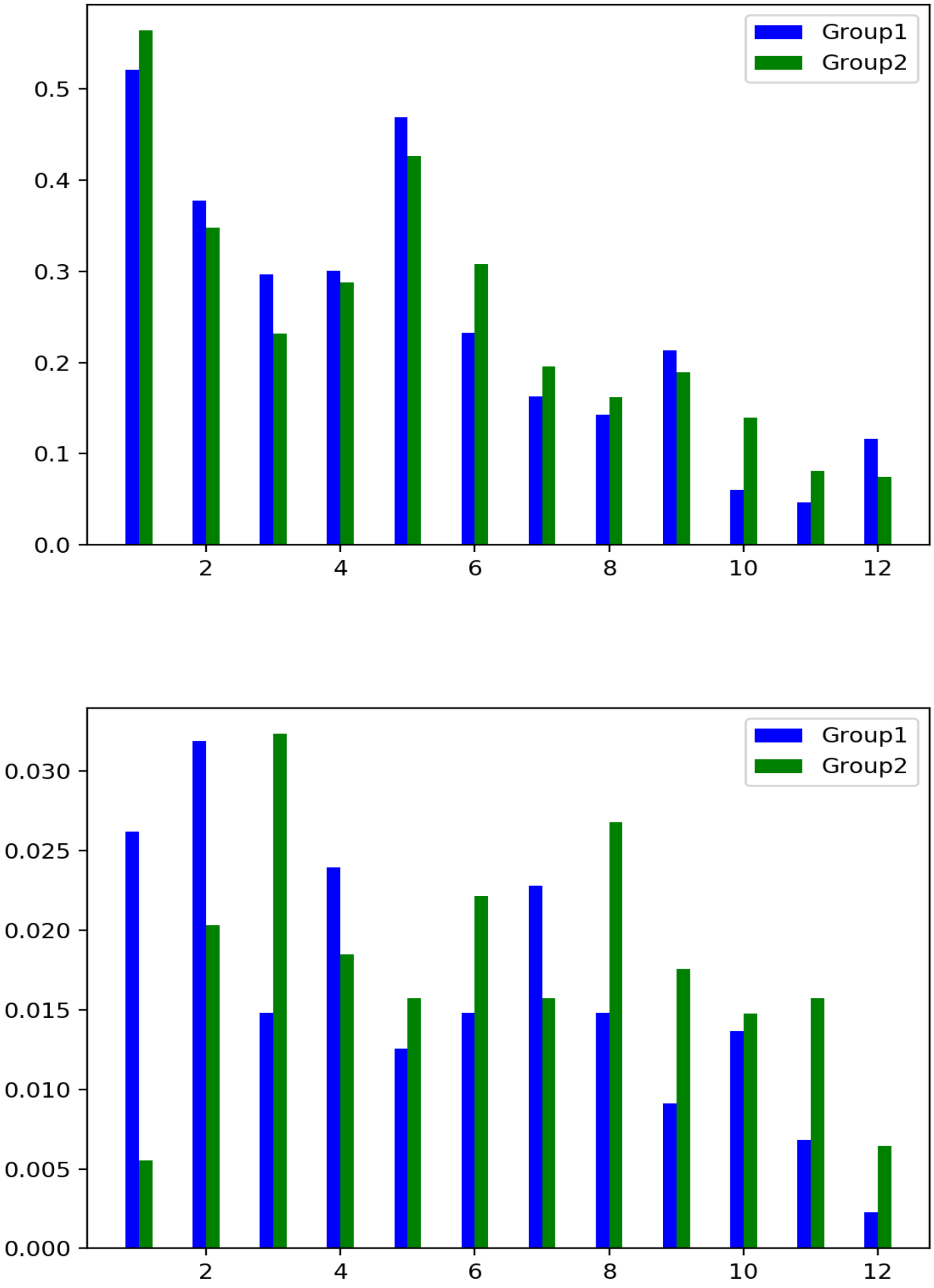
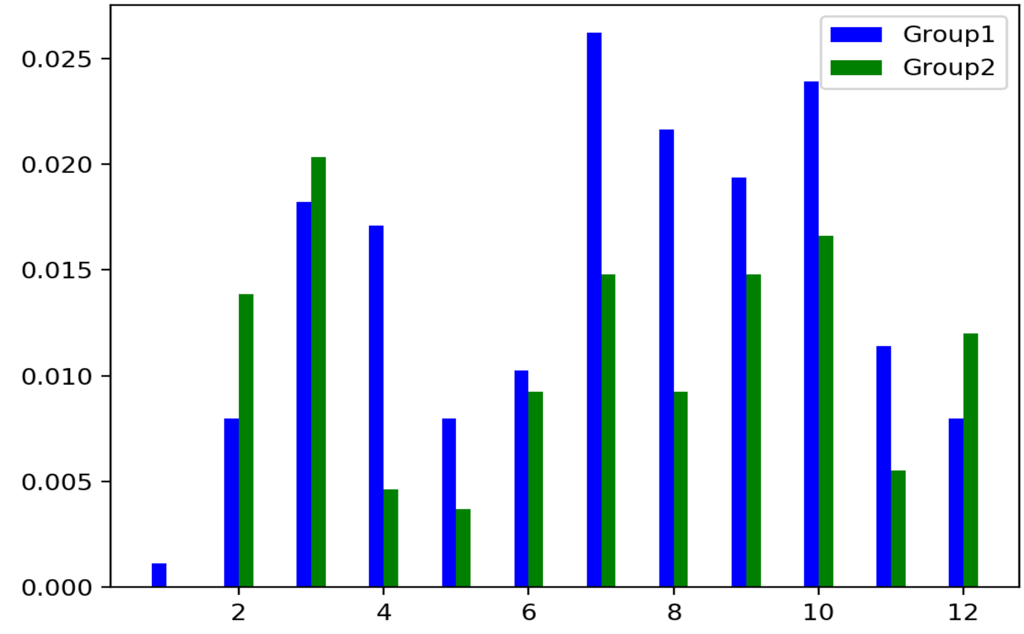


Figure 4.7: Plot for **Thigmotaxis**(top one) and **Incursion** for each trial of all the rodents in group1 and group2



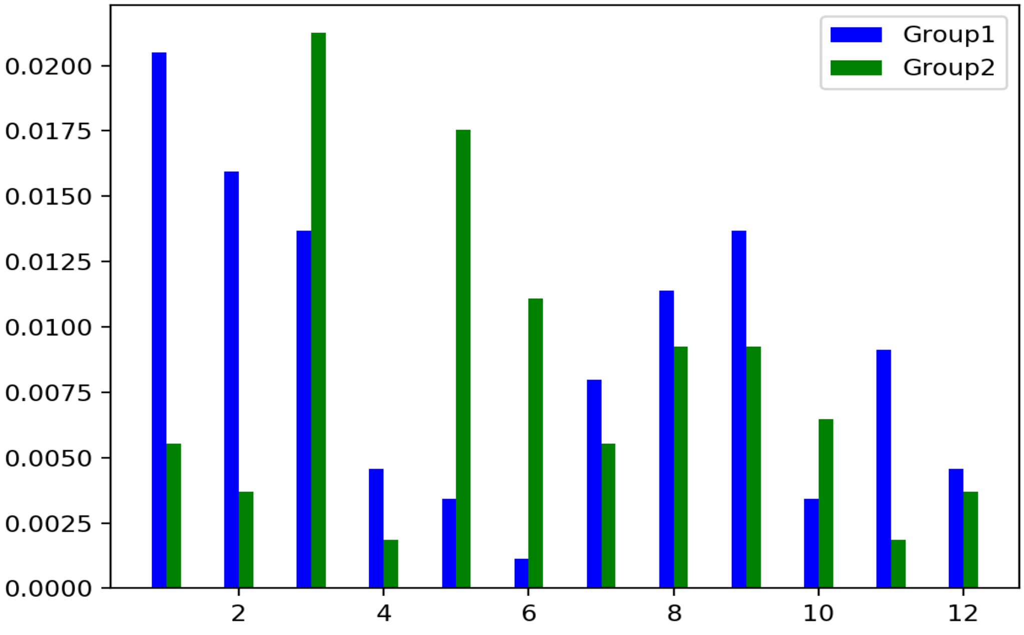


Figure 4.8: Plot for **Scanning**(top one) and **Focused search** for each trial of all the rodents in group1 and group2

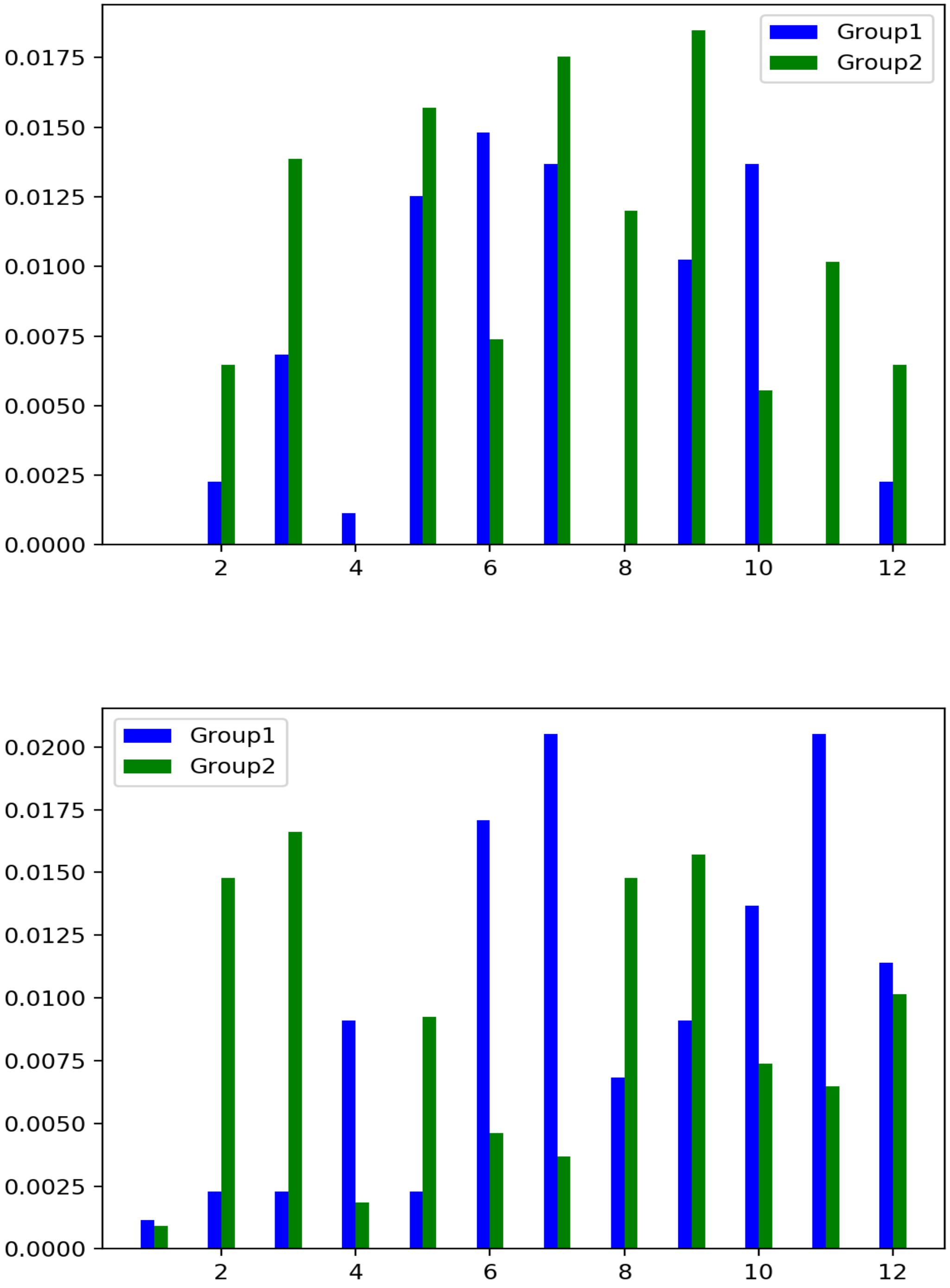


Figure 4.9: Plot for **Chaining response**(top one) and **Self orienting** for each trial of all the rodents in group1 and group2

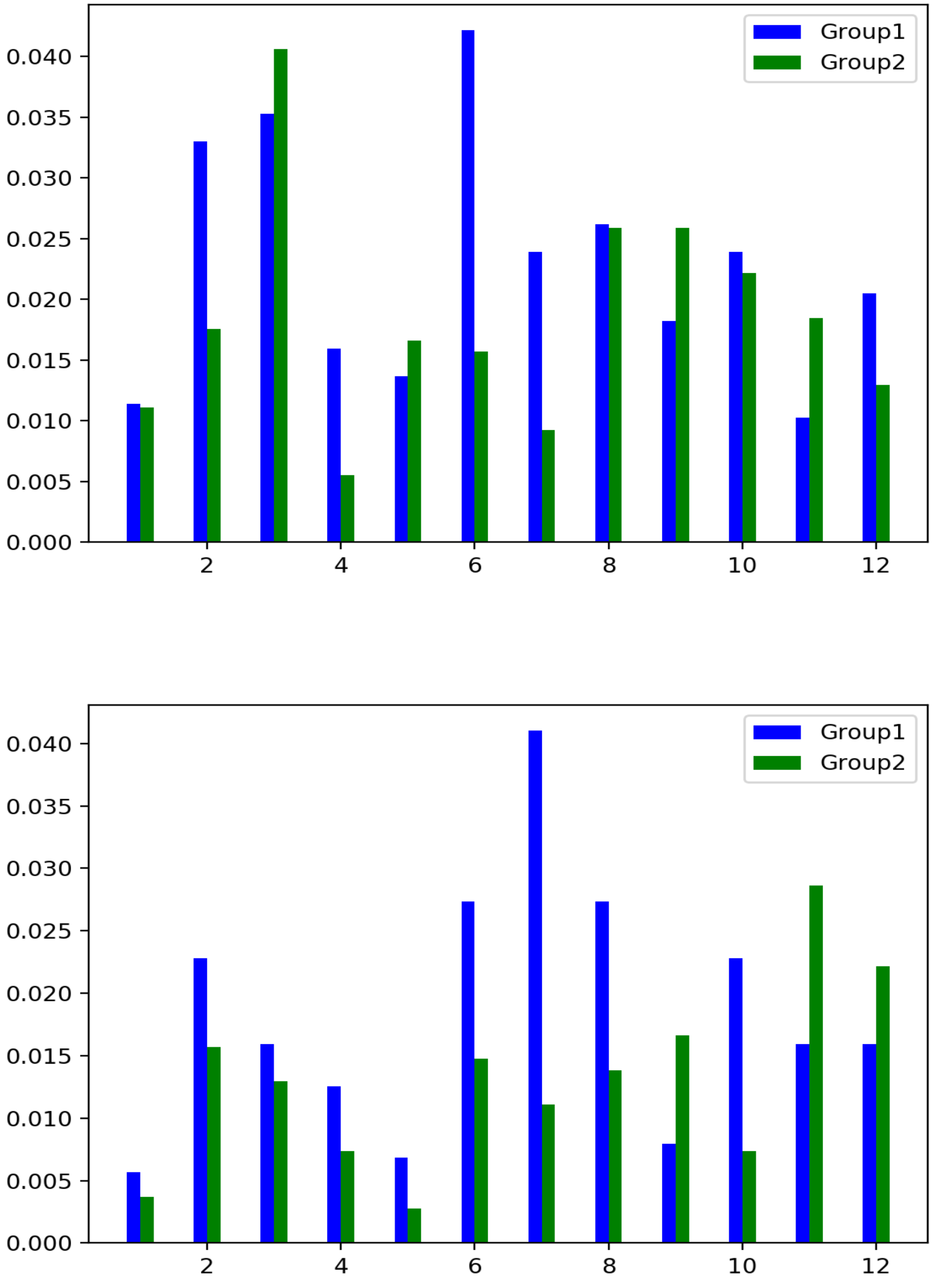


Figure 4.10: Plot for **Scanning surroundings**(top one) and **Target Scanning** for each trial of all the rodents in group1 and group2

Based on the above plots Thigmotaxis is the most used strategy by any rodent in the start. The scale for thigmotaxis is also very high in plot compared to any other strategy. Nest thing to observe is that in most trails in the plot for thigmotaxis group2 rodents have higher values i.e segments of thigmotaxis which mean that group2 rodents spend more times around the walls.

Next Incursion, Scanning surrounding and Target scanning are prominent behaviours. In Incursion for starting trials group1 is in the on with higher values as rodents in this group start making way towards platform in early stages of trials while group2 takes more trials to move away from walls. As for the rest of two strategies both are majorly focused on rodent scanning around the platform or in the quadrant of platform as group1 makes Incursion early on so there values of scanning around the platform is also higher for early trials and group2 have higher values in later trials.

The remaining four strategies don't have significant values as they work primarily in transitioning from one of the four states(Thigmotaxis, Incursion, Scanning surrounding, Target Scanning) to one of the other states from these four states itself.

These all the graphs gives a good understanding of the strategies adopted by rodents and their behaviour traits when trying to reach the platform in stressed environment.

*** All the bar plots in the dissertation are normalized according to this formula.

$$V = V/\sqrt{\sum V^2} \tag{4.4}$$

where **V** is the vector or array we are trying to normalize.

Chapter 5

Conclusion

For understanding the behaviour patterns among rodents when put in stressful environment and comparing the learning rate of two rodent groups, **Control group**(rodent introduced to stress at peripubertal age) and **Stress Group**, this model works with definite efficiency.

The model gives the most prominent strategy adopted by the rodents in starting trails, which is **Thigmotaxis**. Than it gives the most probable state sequences of the rodents trajectory and also gives a definite means to compare the rate of learning among trials. Also we can see how the transition happens from one state to another, which rodent takes which transition sequence thus making their learning rate slower or faster.

Based on the model the learning rate of Control group(Group1) is usually much better compared to Stress Group. This observation can be made in form of their each trial data. Similar observations are made from plots of each rodent individually and Control group rodents usually have faster learning rates. This observation is also made based on the **Transition probabilities**. Observing which group when near the platform or in the state corresponding to the strategy that will eventually get that rodent to platform will deviate from the strategy and choose something arbitrary and unnecessary. By observation this behaviour is exhibited heavily by Stress Group.

This model serves as a good base for the understanding of the behaviour based on the transition probability that determines there next step in the trajectory and also helps in predicting the most probable state sequence.

Chapter 6

Future Work

Based on this model the transition probabilities can further develop to give much better insight on the movement of rodents in stressed environment.

The model can be further developed with more conditions to predict the most accurate trajectory for the movement of rodents.

Using the time constraint properly model can be developed to include spatial factors in segmenting the trajectory.

Chapter 7

Requirements

7.1 Project Requirements

Project requirements are listed in table below. There are three level of priorities: mandatory requirements are the core functionalities that must be implemented, desirable requirements are the functionalities that will be implemented once all the mandatory requirements are implemented and optional requirements are the one that will be additional features in the project. So far only mandatory and optional features are listed below as desirable features will come in action once all the mandatory features are implemented.

ID	Requirement	Priority
1	Understanding of HMM, GHMM, GMM, GM	Mandatory
2	Mathematical interpretation of required machine learning techniques	Mandatory
3	Implementation of Algorithms based on Mathematical techniques	Mandatory
5	Build a GUI software for using this model	Desirable
6	Option for user to add some extra classification features for behavioural traits	Optional

Table 7.1: Project Requirements

7.2 Ethical, Professional and Legal Issues

1. Material(eg code, written information etc.) that are not created by developer will be properly referenced to their main source.

2. Datasets used for testing are provided by the instructor and are referenced is needed.

3. Since there is no human participation or trial, the ethical review is not required.

Bibliography

[1] Aston-Jones, G., Rajkowski, J., and Cohen, J. Locus coeruleus and regulation of behavioral flexibility and attention. In *Progress in brain research*, vol. 126. Elsevier, 2000, pp. 165–182.

[2] Astur, R. S., Ortiz, M. L., and Sutherland, R. J. A characterization of performance by men and women in a virtual morris water task:: A large and reliable sex difference. *Behavioural brain research 93*, 1-2 (1998), 185–190.

[3] Berman, G. J., Choi, D. M., Bialek, W., and Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface 11*, 99 (2014), 20140672.

[4] Bilenko, M., Basu, S., and Mooney, R. J. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning* (2004), ACM, p. 11.

[5] Brandeis, R., Brandys, Y., and Yehuda, S. The use of the morris water maze in the study of memory and learning. *International Journal of Neuroscience 48*, 1-2 (1989), 29–69.

[6] Brown, A. E., Yemini, E. I., Grundy, L. J., Jucikas, T., and Schafer, W. R. A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion. *Proceedings of the National Academy of Sciences 110*, 2 (2013), 791–796.

[7] Buchin, M., Driemel, A., Van Kreveld, M., and Sacristán, V. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science 2011*, 3 (2011), 33–63.

[8] Cornwell, B. R., Johnson, L. L., Holroyd, T., Carver, F. W., and Grillon, C. Human hippocampal and parahippocampal theta during goal-directed spatial navigation predicts performance on a virtual morris water maze. *Journal of Neuroscience 28*, 23 (2008), 5983–5990.

[9] Dalm, S., Grootendorst, J., De Kloet, E. R., and Oitzl, M. S. Quantification of swim patterns in the morris water maze. *Behavior Research Methods, Instruments, & Computers 32*, 1 (2000), 134–139.

[10] Daugherty, A. M., Yuan, P., Dahle, C. L., Bender, A. R., Yang, Y., and Raz, N. Path complexity in virtual water maze navigation: differential associations with age, sex, and regional brain volume. *Cerebral Cortex 25*, 9 (2014), 3122–3131.

[11] Gallagher, M., Burwell, R., and Burchinal, M. R. Severity of spatial learning impairment in aging: development of a learning index for performance in the morris water maze. *Behavioral neuroscience 107*, 4 (1993), 618.

[12] Gehring, T. V., Luksys, G., Sandi, C., and Vasilaki, E. Detailed classification of swimming paths in the morris water maze: multiple strategies within one trial. *Scientific reports 5* (2015), 14562.

[13] Graziano, A., Petrosini, L., and Bartoletti, A. Automatic recognition of explorative strategies in the morris water maze. *Journal of neuroscience methods 130*, 1 (2003), 33–44.

[14] Illouz, T., Madar, R., Louzon, Y., Griffioen, K. J., and Okun, E. Unraveling cognitive traits using the morris water maze unbiased strategy classification (must-c) algorithm. *Brain, behavior, and immunity 52* (2016), 132–144.

[15] Korthauer, L., Nowak, N., Frahmand, M., and Driscoll, I. Cognitive correlates of spatial navigation: Associations between executive functioning and the virtual morris water task. *Behavioural brain research 317* (2017), 470–478.

[16] Lindner, M. D. Reliability, distribution, and validity of age-related cognitive deficits in the morris water maze. *Neurobiology of learning and memory 68*, 3 (1997), 203–220.

[17] Lindner, M. D., and Gribkoff, V. K. Relationship between performance in the morris water task, visual acuity, and thermoregulatory function in aged f-344 rats. *Behavioural brain research 45*, 1 (1991), 45–65.

[18] Luksys, G., Gerstner, W., and Sandi, C. Stress, genotype and norepinephrine in the prediction of mouse behavior using reinforcement learning. *Nature neuroscience 12*, 9 (2009), 1180.

[19] Luksys, G., and Sandi, C. Neural mechanisms and computations underlying stress effects on learning and memory. *Current opinion in neurobiology 21*, 3 (2011), 502–508.

[20] Morris, R. Developments of a water-maze procedure for studying spatial learning in the rat. *Journal of neuroscience methods 11*, 1 (1984), 47–60.

[21] O'keefe, J., and Nadel, L. *The hippocampus as a cognitive map.* Oxford: Clarendon Press, 1978.

[22] Olton, D. S., Becker, J. T., and Handelmann, G. E. Hippocampus, space, and memory. *Behavioral and Brain Sciences 2*, 3 (1979), 313–322.

[23] Petrosini, L., Leggio, M. G., and Molinari, M. The cerebellum in the spatial problem solving: a co-star or a guest star? *Progress in neurobiology 56*, 2 (1998), 191–210.

[24] Piber, D., Schultebraucks, K., Mueller, S. C., Deuter, C. E., Wingenfeld, K., and Otte, C. Mineralocorticoid receptor stimulation effects on spatial memory in healthy young adults: A study using the virtual morris water maze task. *Neurobiology of learning and memory 136* (2016), 139–146.

[25] Schoenfeld, R., Schiffelholz, T., Beyer, C., Leplow, B., and Foreman, N. Variants of the morris water maze task to comparatively assess human and rodent place navigation. *Neurobiology of learning and memory 139* (2017), 117–127.

[26] Vorhees, C. V., and Williams, M. T. Morris water maze: procedures for assessing spatial and related forms of learning and memory. *Nature protocols 1*, 2 (2006), 848.

[27] Vouros, A., Gehring, T. V., Szydlowska, K., Janusz, A., Croucher, M., Lukasiuk, K., Konopka, W., Sandi, C., and Vasilaki, E. A generalised framework for detailed classification of swimming paths inside the morris water maze. *arXiv preprint arXiv:1711.07446* (2017).

[28] Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., Abraira, V. E., Adams, R. P., and Datta, S. R. Mapping sub-second structure in mouse behavior. *Neuron 88*, 6 (2015), 1121–1135.

Appendices

Appendix A

Semi-automated Classification algorithm

The algorithm implemented by *Gehring et al* and *Vouros et al*[12][27] is based on the *Metric Pairwise Constrained K-Means(MPCKMeans)* clustering algorithm implemented initially by *Bilenko et al*[4] in 2004.

MPCKMeans is inspired by the standard K-Means clustering algorithm[**?**] and is considered to be a semi supervised algorithm. In this algorithm the data points are grouped into a particular clusters based on their pattern similarities and are labelled according to their particular behaviour(*must-link and cannot-link*). Moreover this algorithm has the ability to create clusters of different shapes and sizes by using different metrics to minimise the distance between data-points in same clusters and maximise the distance between different clusters.

In this implementation the clusters name are basically the classification of behavioural traits. *MUST-LINK* constraint is generated between two data-points with same label and *CANNOT-LINK* constraint is generated between two data-points with different constraints. Every multi-labelled data-point is considered a distinctive cluster pattern and needs a same multi-labelled data-point for MUST-LINK scenario. Once all the clusters are labelled the classification is established and the results derived based on those classifications.

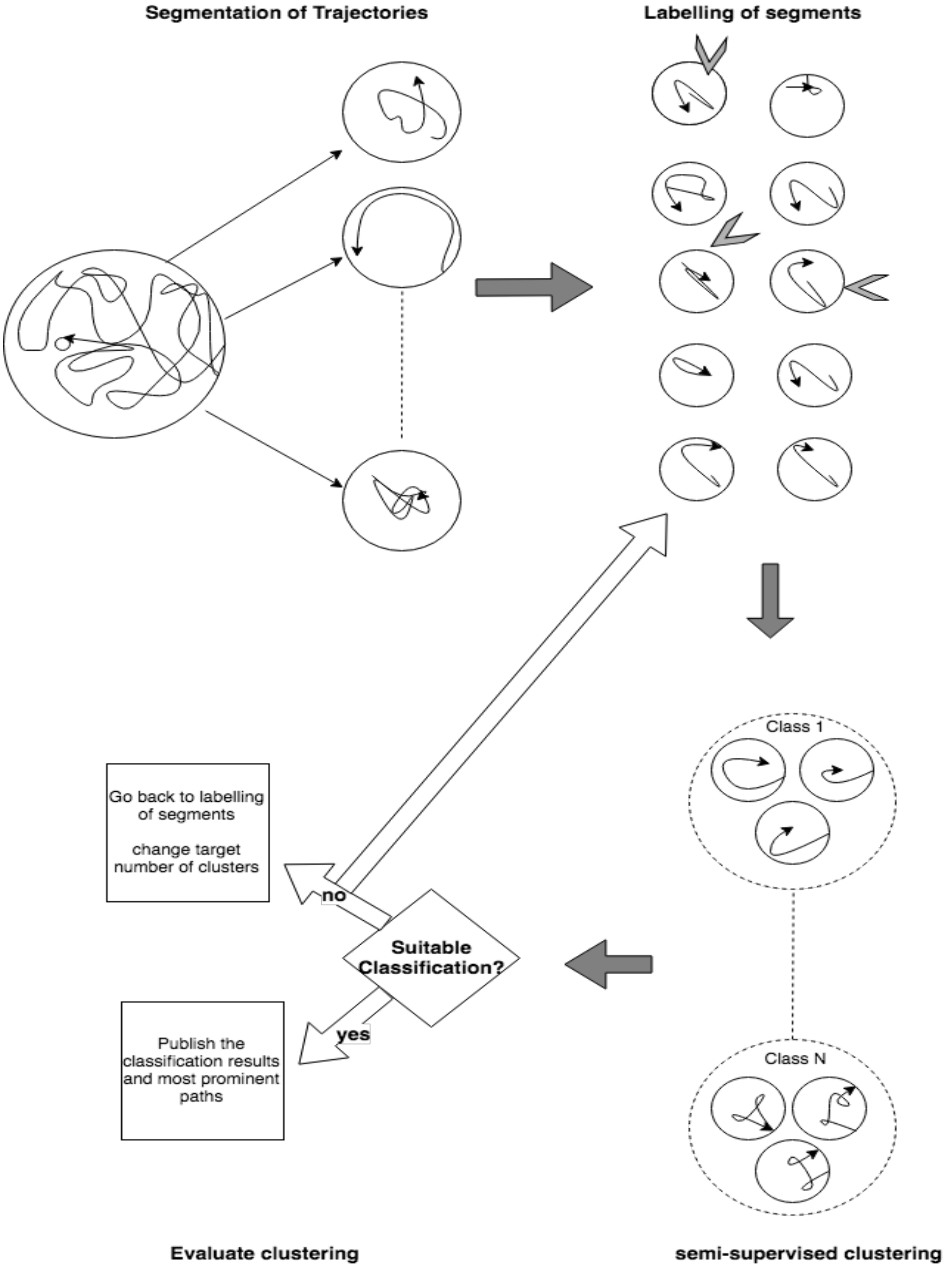

Segmentation of Trajectories
Labelling of segments
Class 1
Class N
Go back to labelling of segments
change target number of clusters
no
Suitable Classification?
yes
Publish the classification results and most prominent paths
Evaluate clustering
semi-supervised clustering

Figure A.1: Diagram showing swimming path classification method using semi-supervised clustering algorithm.

Appendix B

Hidden Markov Model

As the model that will be developed in this project is based on HMM this appendix gives a brief overview of what this algorithm will achieve.

The HMM is a sequence model which allows us to incur sequence of hidden labels from a sequence of observations. HMM models each sequence as a sequence of stationary states,

$$Q = q_1, q_2, ..., q_N \tag{B.1}$$

Each state in the sequence will represent a very small segment of trajectory. An HMM is a Markov chain except the sequence of events in which we are interested is hidden.

B.1 Formal components of HMM

- Set of N states
$$S = s_1, s_2, ...s_N \tag{B.2}$$
- Transition probability where each value represent the probability of moving from state i to j.
$$A = a_{ij} \tag{B.3}$$
- A sequence of T observations.
$$X = x_1, x_2, ..., x_T \tag{B.4}$$
- A sequence of observation likelihood also known as emission probabilities.
$$B = b_i(x_t) \tag{B.5}$$

- Initial probability distribution over states.

$$\pi = \pi_i \tag{B.6}$$

Using these components and fitting them through *Viterbi decoding* we get a best possible state sequence with maximum likelihood of the HMM. In this project the best state sequence will represent the best sequence of segments of trajectory occurrence of which is most likely to happen.

B.2 Viterbi Algorithm

This algorithm consist of four major steps that used the components of the HMM defined above. We will use an array

$$\Psi_t(j) \tag{B.7}$$

to keep track of the best state sequence.

Four steps of Viterbi decoding are:

- **Initialisation :** Delta probability for each state is initialised.

$$\delta_1(j) = \pi_j b_j(x_1),\ \Psi_1(j) = 0,\ 1 \leq j \leq N \tag{B.8}$$

- **Recursion :** For each time scale fo through each state and find the best probability state with best transition probability and than add emission probability to it.

$$\delta_t(j) = max(\delta_{t-1}(j)a_{ij})b_j(x_t) \quad 1 \leq i, j \leq N,\ 2 \leq t \leq T \tag{B.9}$$

$$\Psi_t(j) = argmax(\delta_{t-1}(i)a_{ij}) \quad 1 \leq i, j \leq N,\ 2 \leq t \leq T \tag{B.10}$$

- **Termination :** After the recursion is complete the algorithm terminates with returning values of max likelihood and final best state.

$$P^* = max(\delta_T(i)), \quad q_T^* = argmax(\delta_T(i)) \quad 1 \leq i \leq N \tag{B.11}$$

- **Path Backtracking :** Finally we backtrack through all the states in sequence array and return the best state sequence.

$$q_{t-1}^* = \Psi_t(q_t^*) \quad t = T, T-1, T-2, ..., 2 \tag{B.12}$$

Once this algorithm runs we can get the best state sequence which in turn will be the best classification sequence of behavioural traits label. Based on this classification algorithm we will also get an error rate by using the log likelihood of different HMM's for each trajectory. The comparison can than be established by running the same dataset for GMM algorithm and observe the results.