Unsupervised Learning of Cell Activities in the Associative Cortex of Behaving Monkeys, Using Hidden Markov Models

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

by

Itay Gat

supervised by
Naftali Tishby and Moshe Abeles

Institute of Computer Science and Center for Neural Computation
The Hebrew University of Jerusalem
Jerusalem, Israel.

January 1994
This thesis is dedicated to the memory of my beloved grandmother,
Tova Kahana, who did not live to see its completion.
Acknowledgments

Special thanks are due to:

- Dr. Naftali Tishby, my supervisor, who is in large measure responsible for the completion of this study, for his ideas, continuing interest and enthusiasm.
- Prof. Moshe Abeles, who shared his knowledge unstintingly with me, for his important comments on the work in progress.
- Dr. Hagai Bergman and Dr. Eilon Vaadia for sharing their data with me, and for the numerous stimulating and encouraging discussions of this work.
- Yifat Prut, who gave of her time and helped me through the biological background.
- Iris Haalman and Hamutal Slovin for their valuable assistance in the laboratory.
- Yoram Singer, for his generous advice on mathematical and computational issues.
- My wife, Tamar, who now knows more about behaving monkeys than she ever thought possible, for her encouragement, support, and invaluable editing.

This research was supported in part by a grant from the United States Israeli Binational Science Foundation (BSF).
Contents

I Theoretical Background  
1 Introduction 2 
 1.1 The Cell-Assembly hypothesis 2 
 1.2 Current analysis techniques of the cell assembly hypothesis 3 
    1.2.1 Gravitational clustering 4 
    1.2.2 JPST Histograms 4 
 1.3 Models for simultaneous activity of several cells 5 
    1.3.1 Syn-Fire Chains 5 
    1.3.2 Attractor Neural Network models 5 
 1.4 Statistical modeling 6 
 1.5 Proposed model 7 
    1.5.1 Goals of the model 7 
    1.5.2 Elements of the model 7 
    1.5.3 Our working hypotheses 8 

2 Hidden Markov Models 9 
 2.1 Introduction 9 
 2.2 Definitions 9 
 2.3 Basic problems of the HMM 12 
 2.4 Solution to the evaluation problem 12 
 2.5 Solution to the decoding problem 13 
 2.6 Analysis of the learning problem 14 
    2.6.1 The Baum-Welch algorithm solution 15 
    2.6.2 The segmental K-means algorithm 17 
 2.7 Clustering techniques used in HMM 17 
    2.7.1 KL-divergence 18 
    2.7.2 Characteristics of the KL divergence 21 
 2.8 Hidden Markov Model of multivariate Poisson probability observation 22 
    2.8.1 Clustering data of multivariate Poisson probability 22 
    2.8.2 Maximum likelihood of multivariate Poisson process 23 

II Model Implementation 25
3 Physiological Data and Preliminary Analysis
   3.1 Frontal cortex as a source of data ................. 26
   3.2 Description of physiological experiment ............. 27
      3.2.1 Behavioral modes of the monkey ................. 27
      3.2.2 Data from recording session .................. 27
   3.3 Basic tools of analysis ............................ 28
      3.3.1 Dot display tool ............................... 28
      3.3.2 Temporal cross correlation tool ................ 29
   3.4 Quality of cells recorded ........................... 31
      3.4.1 Isolation score given to cells during recording session .. 31
      3.4.2 Stationarity of a cell .......................... 32
      3.4.3 Firing rate of cells as a quality measure .......... 33
      3.4.4 Responsiveness of cells to events ............... 34

4 Application of the Model ............................... 36
   4.1 Data selection stage ............................... 36
   4.2 Computation of parameters used in the model .......... 37
   4.3 Clustering of data vectors ........................... 38
   4.4 Implementation of a full HMM ......................... 39
   4.5 Insertion of firing rate cross-correlations into the model ... 40
   4.6 Search for correlation between state sequences and events ...... 42
   4.7 Display tools ........................................ 42
   4.8 Temporal cross-correlation in different states .......... 43

III Results .................................................. 44

5 Temporal Segmentation .................................... 45
   5.1 Multi-cell activity in different states ............... 45
   5.2 Consistency of clustering results .................... 45
   5.3 Coherency of the clustering results .................. 47
   5.4 States’ length distribution ........................... 49
   5.5 Effect of full HMM implementation .................... 51
      5.5.1 Differences between the Baum-Welch algorithm and the segmental K-means .................. 51
      5.5.2 Changes in temporal segmentation due to a full HMM 52
   5.6 Effect of insertion of firing rate cross-correlations into the model .................. 53
   5.7 Prediction of events based on state sequences .......... 54

6 Temporal Cross Correlations in different states ......... 59
   6.1 Working hypotheses ................................... 59
   6.2 Cells with sufficient data ............................ 59
   6.3 Flat correlograms vs. functional connectivity ........ 63
   6.4 Types of functional connectivity ........................ 66
6.5 Modification of pair-wise cross-correlation functions between states

6.6 Contamination of spike train as a cause of MPCS
   6.6.1 Insufficient isolation between cells
   6.6.2 Cells recorded from the same electrode

6.7 Types of MPCS

6.8 Changes in firing rates of MPCS

IV Discussion

7 Discussion

7.1 Hypotheses of the research

7.2 Interpretation of results
   7.2.1 Consistency of the model
   7.2.2 Multi-dimensionality of the model
   7.2.3 Differences between behavioral modes
   7.2.4 Different models used in the research
   7.2.5 Prediction of events by the model
   7.2.6 Changes in functional connectivity between states

7.3 Limitations of the model

7.4 Conclusions and suggested further work
## List of Figures

2.1 A simple Hidden Markov Model with 3 states. .................. 10

3.1 A block diagram of the behavioral modes of the monkey. ... 28
3.2 An example of a dot-display. ........................................ 30
3.3 An example of a cross-correlation function (correlogram) of
two cells. ................................................................. 31
3.4 An example of nonstationarity in the firing rate of a cell, in
which three distinct sections (marked in the left side of the
dot-display) can be seen. ............................................. 33
3.5 An example of different firing rates of two cells from the same
recording session. ..................................................... 34
3.6 An example of two cells that show different responses to the
same stimulus. ............................................................ 35

4.1 An example of 8 spike trains of simultaneously recorded cells,
with a short time firing rate. ......................................... 38
4.2 An example of execution of a K-means algorithm for different
number of states, using the same data. .......................... 40
4.3 An example of execution of a K-means algorithm for two sets
of data vectors. ........................................................... 41

5.1 The centroids of all the states. Each line represents the cen-
troid of one state, and each bar on the line represents the
firing rate of one cell in that centroid. ......................... 46
5.2 An example of three sets of clustering results. The upper sets
are from the same behavioral mode, the lower set is from a
different behavioral mode. ....................................... 47
5.3 An example of the coherency measure difference between two
behavioral modes, in different starting conditions. ........ 49
5.4 An example of length distribution among segmentation of dif-
ferent behavioral mode. ............................................. 50
5.5 An example of a typical segmentation of spike trains of 8 cells. 51
5.6 An example of two temporal segmentations of the same data
(a.) and all their clusters’ respective probabilities (b.) .... 56
5.7 An example of covariance matrices of 7 cells in 3 different states. 57
5.8 An example of two segmentations and two labelings of spike trains at different time points of the same recording session. 58

6.1 Two cross correlograms that show the effects of insufficient data. 60

6.2 An example of a cross-correlogram without sufficient data but with clear functional connectivity 62

6.3 An example of three correlograms. a: flat correlation function, b: nonzero correlation, c: questionable nonzero correlation 63

6.4 An example of a correlogram which is basically flat although the cross-correlation function crosses the significance lines 64

6.5 An example of different types of functional connectivity 67

6.6 All the correlograms of a pair of cells in different states 68

6.7 An example of 6 correlograms showing different functional connectivities 72

6.8 An example of wide to narrow functional connectivities 73

6.9 An example of inhibitory to flat functional connectivities 73

6.10 An example of narrow to flat functional connectivity 74

6.11 An example of one sided to double sided functional connectivities 76

6.12 An example of a dramatic change in firing rate without any change in the functional connectivity 76

7.1 An example of HMM combined from simple task specific models, and the embedding left-to-right model 85
List of Tables

2.1 Simulated execution of the HMM shown in figure 2.1 .... 11

4.1 Presentation of the recording sessions used throughout the experiments. For each recording session, the number of well-isolated cells used is indicated, as is the total length of the session (in seconds) and the cortical area from which the data were recorded. These time tables are presented separately for the “Go” mode and the “No-Go” mode. ................. 37

5.1 The coherency of the best model for the “Go” mode and the “No-Go” mode at each of the recording sessions. ............ 48

5.2 The percentage of segments with lengths up to 1500 ms, for both behavioral modes in all the recording sessions. ...... 52

5.3 The likelihood values achieved in the different training algorithms. ................................................................. 52

5.4 Typical examples of prediction of events using states’ sequences. 54

6.1 The percentage of data used for the temporal cross-correlation, from the total length of recorded data. ............... 60

6.2 The number of sufficient data cross-correlograms, among the cross-correlograms of the chosen cells. ............ 62

6.3 The number of non-flat correlograms among the sufficient data correlogram .................................................. 65

6.4 The classification of non-flat correlograms into four types of functional connectivity ............................ 66

6.5 The number of MPCS occurrences - classified into “Go” trials and “No-Go” trials, and compared with the total number of sufficient data cell-pairs. ......................... 69

6.6 Classification of MPCS occurrences for each recording session into three groups: A or B types, B or C types, and all the other types. ................................. 70

6.7 The number of MPCS occurrences from the same electrode and from different electrodes, in the 2 behavioral modes. ... 71

6.8 Comparison between changes in firing rate vs. changes in MPCS. ................................................................. 75
Abstract

Hebb hypothesized in 1949 that the basic information processing unit in the cortex is a cell-assembly which can act, briefly, as a closed system after stimulation has ceased, and constitutes the simplest instance of a representative process. This hypothesized cell-assembly may include thousands of cells in a highly interconnected network. The cell-assembly hypothesis shifts the focus from the single cell to the complete network activity. So far, there has been no general method for relating extracellular electrophysiological measured activity of neurons in the associative cortex to the underlying network or the cell-assembly states. It is proposed here to model such data as a pair-wise correlated multivariate Poisson process. Based on these parameters, a Hidden Markov Model was computed. This modeling yielded a temporal segmentation and labeling of the data into a sequence of states.

The first hypothesis of this work was that a connection exists between the states of the model and the behavioral events of the animal. i.e. based on the sequence of states of the model, the observed actions of the animal could be predicted. The second hypothesis was that a connection exists between the states defined above and the functional interaction between cells, i.e. the functional interaction between pairs of cells changes in different cognitive states.

The application of this approach was demonstrated for temporal segmentation of the firing patterns, and for characterization of the cortical responses to external stimuli. This modeling was applied to 6 recording sessions of several single-unit recordings from behaving monkeys. At each session, 6-8 single-unit spike trains were recorded simultaneously. Using the Hidden Markov Model, two behavioral modes of the monkey were significantly discriminated. The two behavioral modes were characterized by different firing patterns, as well as by the level of coherency of their multi-unit firing activity. The result of the modeling showed a high degree of consistency, which implies that the model succeeds in capturing a basic structure underlying the data. Significant changes were found in the temporal cross-correlation of the same pair of cells in different states, indicating different functional connectivities of the small network being recorded. These changes suggest that the modeling captures the activity of the network and that the states of the model can be related to the cognitive states of the cortex.
Part I

Theoretical Background
Chapter 1

Introduction

1.1 The Cell-Assembly hypothesis

The cortex is considered the part of the human brain that is essential to the processing of information. The human cortex consists of a vast number of neurons (around $10^{10}$), which are connected via synapses, each neuron typically receiving $10^4 - 10^5$ synapses [1]. It is assumed that this structure enables the brain to perform the complicated actions it is capable of. The basic question emerging from this structure concerns its relationship to the complex action for which it is responsible.

One of the common hypotheses related to this question is the Cell-Assembly Hypothesis suggested by Hebb in 1949. “It is proposed first that a repeated stimulation of specific receptors will lead slowly to the formation of an ‘assembly’ of association-area cells, which can act briefly as a closed system after stimulation has ceased; this prolongs the time during which the structural changes of learning can occur and constitutes the simplest instance of a representative process (image or idea)” [36] This hypothesized cell-assembly may include thousands of cells in a highly interconnected network. The cell-assembly is further characterized by stronger than average connections between the cells which comprise it. The organization of the cell-assembly is achieved via these strong connections, i.e. there is a mechanism that creates these assemblies by strengthening the connection between cells. Hebb further hypothesized that this mechanism is related to the ordered activation of cells, that is, whenever two cells are activated one after the other, the connection between them is reinforced. The cell-assembly is a group of interconnected cells, whose participation in the cell-assembly is not determined by proximity. One cell may participate in several cell-assemblies, and it is possible for cells which are next to each other not to be part of the same cell-assembly. The definition of a cell-assembly is, therefore, more of a functional than an anatomical one, as the cell-assembly is not defined by specific physiological locations. The information processing in the cortex is dynamic in time, and tends to alternate between different activation states of one cell-assembly, and between activations of several cell-assemblies. The same cell-assembly may be utilized in different computational tasks. Thus, a typical computation process consists of a sequence of these functional units (cell-assemblies), and will henceforth be referred to as a sequence of cognitive states.

Defining the cell-assembly as the basic unit of processing suggests the notion of the statistical behavior of a single cell in that network: “in a larger system a statistical constancy might be quite predictable” [36]. This statistical behavior is also expected due to the fact that a single cell may
CHAPTER 1. INTRODUCTION

participate in a number of assemblies. The notion of the cell-assembly has been formalized since Hebb first suggested it \[72, 54\]. Furthermore, the cell-assembly hypothesis matches the up-to-date knowledge about the anatomical structure of the cortex, especially the structure of high cortical areas - the associative cortex \[31\].

1.2 Current analysis techniques of the cell assembly hypothesis

One of the traditional views of the neuronal activity stressed the importance of the activity of single cells \[13\]. Methods based on this view show quite remarkable results in many cortical and sub-cortical areas. These results include relationships between highly specific changes of firing rates and sensory or motor processing, or even cognitive and other higher brain function such as perceptual analysis of sensory input, selective attention, short time memory, etc. \[23, 53, 35\]. The usual way of computing the activity of a single cell is by the peri-stimulus-time (PST) histogram, which is an average measure of the firing rate probability relative to the stimulus presented. In high cortical areas it was found that in order to attain a reasonably smoothed PST it was necessary to average the activity over many repeated activations (typically at least 30 to 50). This fact by itself shows that the requisite information cannot be coded in the activity of a single cell alone \[65\]. Still, the PST histogram is a valuable tool for measuring the firing rate of cells, and can give a flexible estimation in the time domain.

The cell-assembly hypothesis shifts the focus from the single neuron to the complete network activity. This view has led several laboratories to develop technology for simultaneous multi-cellular recording from a small region in the associative cortex \[1, 43, 44\]. Extracellular electrophysiological measurements have so far obtained simultaneous recordings from just a few randomly selected neurons (about 10) in the cortex, a negligible number compared to the size of the hypothesized cell-assembly. These are recordings of the simultaneous firing activities of the randomly selected neurons of a *behaving animal*. *Behaving animal* is the term used to describe a conscious animal engaged in a specific, pre-defined activity. The discrete action potentials shall be called henceforth the *spike trains* of cells.

In order to process this multi-cellular recording information, and to relate the data to the hypothesized cell-assembly, several techniques have been developed. The best known of them involves computing the pair-wise cross-correlation function between cells \[57, 34, 55\]. This technique analyzes the spike trains of a pair of cells for possible temporal relations, by computing the cross-correlation function of the two spike trains. Departures from background in the correlogram are taken as an indication of *functional connectivity* between the two cells \[8, 69\]. The term *functional connectivity* (also referred to as *effective connectivity*) \[50, 57\] is used in this context because of the cell-assembly hypothesis which implies that this correlation is due to the activity of the whole cell-assembly.

This analysis method suffers from some serious limitations:

1. The number of cross-correlation functions grows quadratically with the number of recorded cells: for $N$ cells one has $N(N-1)/2$ different pairs. This number is further multiplied by the number of different conditions under which these cells are checked (different stimuli, different behavioral modes of the animal, etc.). This growth makes the method inapplicable even for a small number of cells.
2. The difficulty of monitoring the dynamics of cross-correlation over time.

3. The difficulty of processing information from more than two or three cells simultaneously.

4. The need for long stationary recordings that, presumably, originate from the same activation of the cell-assembly.

The last point raises the issue of time in this analysis method. Not only does the cross-correlation analysis require long time intervals of spike trains, it also necessitates the repetitive measurements of these same intervals in order to reach significant results. The use of repetitive measurements rests on the assumption that under similar external circumstances, the recorded cells behave in a similar way. This assumption ignores the following facts:

- The recorded cells are randomly selected from a much larger network. As a result of this, and of the statistical nature of their behavior, single cells may function in different ways at different points of time, while the whole network acts in a consistent manner.

- The ability to produce exactly the same external circumstances in each repetition is very limited.

There are several methods that attempt to overcome some of these limitations. Most notable among them are the Gravitational Clustering [32, 31], and the Joint Peri Stimulus Time Histogram (JPST)[7, 8].

1.2.1 Gravitational clustering

Gravitational clustering attempts to overcome the enormous proliferation of cross-correlation functions to be examined, and transform them into a single computation. This transformation is carried out by modeling each of the $N$ neurons as particles in an $N$ space. All particles are initially equidistant. Each particle carries a charge which is incremented with every firing of the corresponding cell, and then decays with an appropriate time constant. These particles attract each other with a force proportional to the product of their charges. Thus they move towards each other with a velocity proportional to the degree of firing correlation. This system converges into clusters of highly correlated cells (if such exist). The clustering process over time, and the clusters formed, tell us about the nature of these correlations. This computation is particularly effective in screening large amounts of data, and identifying the spike trains that should undergo a more detailed inspection. Considerably more detailed description of the gravity transformation can be found in [30, 7].

The main problem of this method is the cross talk between cells. That is, a highly correlated activity of two cells alters the distance of these cells from the other cells. This change of distance is not the result of correlated activity between these two cells and other cells. Another problem is that this method is not sensitive to high order correlations between cells.

1.2.2 JPST Histograms

The Joint Peri Stimulus Time Histogram attempts to overcome the problem of dynamics found in the cross-correlation between cells. This is done by drawing a scatter diagram of the co-firing of two cells, time-locked to the occurrence of a certain event, and counting the number of joint
occurrences in a cartesian grid of bins. The output of this process is a matrix of counts $M$, in which each element $M_{ij}$ is the average co-firing of the two cells within a specific time interval (the $i$th bin for the first cell and $j$th bin for the second cell), time-locked to the event used as a trigger. Further normalization is applied to this matrix on the basis of the expected amount of co-firing (computed from each neuron PST). The normalized matrix provides evidence for the dynamics of the functional connectivity between the neurons, relative to the event used.

This method is appropriate for studying the connection between two cells, but suffers from the same problems as the cross-correlation technique, regarding the large number of graphs and the time needed to scan and process them.

1.3 Models for simultaneous activity of several cells

Several models and hypotheses followed Hebb’s ideas. Two of them are described here: the Syn-Fire Chain [3, 2, 6], and the Attractor Neural Network [37, 11]. These models assume that “learning” is based on selective modification of the connectivity among neurons, and “computations” are carried out by selective activation of interconnected neurons. These models do not contradict each other, and can be viewed as different aspects of the same physical system.

1.3.1 Syn-Fire Chains

The Syn-Fire Chain model is based on synchronous temporal structures of spike trains. According to this model, the activity in the cortex is propagated in a network made of a chain of diverging-converging links, where each of these links is made of several neurons. The information in this model is propagated in the chain by secure transmission of a synchronous volley from link to link. This model, like the cell-assembly, takes not only an anatomical form, but also a functional one, i.e. the neurons participating in the Syn-Fire Chain may be scattered and intermingled with each other and with many other neurons. Furthermore, the same neuron may participate in several syn-fire chains. This model predicts the existence of temporal structures in the spike trains of cells, and analyzes them [3, 5].

1.3.2 Attractor Neural Network models

In the last decade there has been impressive development in the field of physical models of neuron-like networks, many of which can be analyzed by statistical mechanics, via the analog with spin-glass systems. One of the notable models is the Attractor Neural Network (ANN). This model presents a network consisting of a large number of simple look-alike units, that imitate some of the basic functions of real neural networks. Each of these units is connected via weighted links to some, or all, of the other units, and alters its state according to these links and to its current state. These units are, therefore, referred to as neurons, and the whole system is referred to as a neural network.

This description defines a dynamical system, which for any given initial set of neuron states, goes on wandering among the $2^N$ possible configurations of the network, where $N$ is the number of neurons. The basic concept in this framework is the attractor. An attractor is defined as a state (or a cycle of states) of the network that is reached through the dynamics of the system, from different initial configurations. Reaching an attractor in the network can be interpreted as recalling a pattern that is stored in memory. The attractors reached can be static ones (an example
of this type can be seen in the Hopfield model [37]. In a more general setup, cyclic attractors may appear.

This description suggests that an ANN is capable of performing memory recall by using the attractors of the network [41]. This ability may be seen as the first step towards a computing network that dynamically goes through a sequence of alternating attractors. This sequence of attractors may be related to the ideas concerning the cell-assembly hypothesis mentioned above [42].

1.4 Statistical modeling

One of the basic tools used in this work in order to combine the information about cell-assemblies and the recording from behaving animals, is the statistical modeling [22]. The basic task to be carried out on the data is to “model” it, i.e. to be able to capture the inherent structure of information it contains. This ”capturing” would allow the model to analyze the information in the data, and to make predictions on the basis of what has been modeled.

In the field of learning theory, a separation exists between algorithmic procedures that use labeled samples for discrimination and are said to be supervised, and procedures that use unlabeled samples and are called unsupervised. All the methods of analyzing the data collected from the cortex, described above, were processed in a supervised manner. The labeling used in these processes was the timing of external events in the experiment.

Unsupervised modeling has both benefits and disadvantages in the modeling of cortex activity, when compared to supervised modeling. The basic advantage of this method is that it enables the direct modeling of the data without making assumptions as to the nature of the data. For example, the modeling techniques shown above assume a direct connection between events of the experiment (stimuli to the animal, or behavioral actions) and the sequence of cognitive states the cortex goes through. This assumption may not always hold, as explained before. An attempt to model that data in an unsupervised manner, on the other hand, may reveal a direct connection between the spike trains of several cells and the cognitive states of the cortex. The clear disadvantage of unsupervised learning is that only a collection of unclassified examples exists - so the basic concepts are not known in advance.

In this work we will focus on one type of unsupervised learning procedure. This type assumes that the data is unlabeled, but that the class of functions of the underlying probabilities producing these samples is known. An example of this type is the classification of samples received from a mixture of two Gaussians with different means and variances. The statistical modeling in this case is required to find a way to estimate the parameters of the probability function. In the Gaussian mixture case, these are the means and variances of the processes.

While this research was in progress, another work in the same field was begun in our laboratory [64]. In this work the same data was modeled using a HMM, but with two important differences:

1. The model was built in a supervised manner, using only the spike trains of the first 4 seconds after a pre-defined external stimulus.

2. The observations of the model were computed every ms, and included the information as to which cell emitted a spike. The possibility that two or more cells emitted spikes was disregarded.
CHAPTER 1. INTRODUCTION

This data used by the model was divided into four different groups (according to the type of stimulus presented to the animal), and after appropriate training the model was able to predict the stimulus from the spike trains. While writing this report, another relevant work was brought to our attention [63]. The recording in this work were taken from the visual cortex of anesthetized monkeys. In this work another supervised HMM analysis of spike trains was carried out. The results of this work show the possibility of predicting the different visual stimuli presented to the anesthetized animal from the recorded spike trains.

1.5 Proposed model

1.5.1 Goals of the model

The aim of the work presented here is to find a connection between the hypothesized cell-assemblies and the attractors introduced in ANN, and the spike trains of several cells recorded simultaneously. It is related to a core physiological question: whether the computations in the brain are carried out by cell-assembly-like circuits, i.e. by transitions between distinct states. The time constants of these processes are assumed to be in the order of tens to hundreds of ms, and these are the time constants that will be used in the model.

The model presented tries to overcome some of the problems of the current analysis techniques described above, in the following ways:

- The relation between the states and the parameters of the model based on the spike trains is a direct one, i.e. those spike trains are the stochastic output of the computed states.
- The estimation of the states is based on stochastic point-wise processes, i.e. repetitions of intervals of the data are not averaged.
- The computations do not make use of the events marked on the spike trains (such as the stimulus to the animal or the behavioral reactions).
- The computation is based on a truly vectorial model which uses all the parameters computed from the spike trains simultaneously, i.e. computation is carried out in an $N$ dimensional space, where $N$ is the number of cells used.

1.5.2 Elements of the model

The analysis method presented in this work is based on a model with the following assumptions:

- The spike trains are produced as a stochastic output from a Markov process.
- The states of the Markov process change according to specific transition probabilities.
- The states cannot be observed directly. Only the stochastic output of the state at each time is observable.

One type of such a process is the first order Hidden Markov Model (HMM) presented in Chapter 2 in detail.
In order to model the spike trains of cells with a HMM, the architecture of that model must be chosen, and its parameters estimated [29]. In this realization of the model, the simultaneous firing rates of the cells, and the pair-wise cross-correlation between them, were used as the statistical measure of the states. The firing rates were modeled as a multivariate Poisson process. Based on these estimated parameters, a Hidden Markov Model was computed. This modeling also yielded a temporal segmentation of the data into sequences of states.

### 1.5.3 Our working hypotheses

The first hypothesis of the model is that a connection exists between the states of the model and the behavioral events of the animal. This connection may be found in a correspondence of the model to some specific neural activity. It may also be found in the ability to predict observed actions based on the sequence of states of the model. The states of the model estimated from the data are assumed to be connected to a sequence of cognitive states related to the action taken by the animal. The purpose of this work is to substantiate this relationship in a scientific qualitative manner. In order to check this hypothesis, different aspects of the temporal segmentation of the data will be checked. These aspects include:

- The examination of state representatives on the basis of their mutual activity.
- The search for differences between behavioral modes.
- The coherency of the model, i.e. how accurately does the model describe the data.
- The ability of the model to predict events from the states (or sequences of states).

The second hypothesis is that a connection exists between the states defined above and the functional connectivity between cells, i.e. the functional connectivity between pairs of cells changes in different cognitive states. In order to check this hypothesis, the temporal cross-correlation functions between the same pair of cells will be computed for different states of the model. These functions will be compared, and changes between states will be sought. If the model does indeed captures some notion of states’ activity in the cortex, the cross-correlations are expected to show different types of functional connectivity in different states.
Chapter 2

Hidden Markov Models

2.1 Introduction

Hidden Markov Modeling (HMM) is a probabilistic technique for the study of time series [58, 19, 60]. Hidden Markov theory permits modeling with many of the classical probability distributions. The costs of implementation are linear in the length of the data. These and other desirable features have made Hidden Markov methods increasingly attractive for tackling problems in language, speech and signal processing.

Although initially introduced and studied in the late 1960s and early 1970s, Hidden Markov Modeling has become increasingly popular in the last several years. There are two strong reasons why this has occurred. First, the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Second, the models, when applied properly, work very well in practice for several important applications [46, 61].

2.2 Definitions

The basic concepts of the Hidden Markov Models are as follows:

Given a system which may be described at any time as being in one of a set of $N$ distinct states, $S_1, S_2, \cdots, S_N$, as illustrated in figure 2.1 (where $N = 3$ for simplicity). At regularly spaced discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. These probabilities are denoted in the figure by $a(i, j)$ variables. The actual sequence of states would be denoted as $q_t, 1 \leq t \leq T$, where $T$ is the number of discrete steps the systems undergoes. A full probabilistic description of the above system would, in general, require specification of the current state (at time $t$) as well as all the preceding states. For a special case of a first order Markov chain, this probabilistic description is truncated to just the current and predecessor state. Furthermore, the only considered processes are the ones in which the probability is stationary, i.e. the probabilities $a_{i,j}$ are independent of time. In the example shown in figure 2.1 all the transitions between states are allowed, i.e. are with a non-zero probability. This is a special case of Markov models in which ergodicity is kept. There exist also non-ergodic models in which some of the transition probabilities are not allowed (set to zero), thus producing a specific structure of states.

The above stochastic process could be called an observable Markov chain since the output of
the process is a set of states at each instant of time, where each state corresponds to a physical event.

In the Hidden Markov Models, the model described above is extended to include the case where the observation is a probabilistic function of the state. The resulting model is a double embedded stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce the sequence of observations. These observations may be discrete symbols from a finite alphabet, or continuous variables from a known distribution.

This double-embedded stochastic process is shown in figure 2.1 in the following manner: the states are shown to be hidden by the dotted horizontal line presented in the upper part of the figure, where the observation of each state is shown as a dotted arrow emerging from the state and crossing that horizontal line.

The formal constituents of the discrete HMM are as follows:

1. $N$, The number of states in the model.
2. $M$, The number of distinct observation symbols per state, i.e., the discrete alphabet size. This description will be altered later to include the continuous distributions.


5. $\pi$, The initial state distribution vector.

Given appropriate values of $N, M, A, B$ and $\pi$, the HMM can be used as a generator to give an observation sequence

$$O = O_1 O_2 O_3 \cdots O_T,$$

where each observation $O_t$ is one of the symbols, and $T$ is the number of observations in the sequence. The symbols are denoted as $V = v_1, v_2, \cdots v_M$ in the discrete case. The execution of the HMM generator is as follows:

1. Choose an initial state $q_1 = S_i$ according to the initial state distribution $\pi$.

2. Set $t = 1$.

3. Choose $O_t = v_k$ according to the symbol probability distribution in state $S_i$, i.e. $b_j(k)$.

4. Transit to a new state $q_{t+1} = S_j$ according to the state transition probability distribution for state $S_i$, i.e. $a_{ij}$.

5. if $t = T$ stop.

6. $t \rightarrow t + 1$

7. Go to step 3.

An example of this process may be seen in table 2.1. This example is related to the Hidden Markov Model presented in figure 2.1. In this example, the size of the alphabet is 4, thus 4 different symbols may be emitted from the states. The hidden states are presented in the top line and the observations produced by them are presented in the bottom line.

<table>
<thead>
<tr>
<th>States</th>
<th>$q_1$</th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_2$</th>
<th>$q_2$</th>
<th>$q_1$</th>
<th>$q_3$</th>
<th>$q_2$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbols</td>
<td>$v_3$</td>
<td>$v_3$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_3$</td>
<td>$v_4$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_4$</td>
</tr>
</tbody>
</table>

Table 2.1: Simulated execution of the HMM shown in figure 2.1

The above procedure can be seen as both a generator of observations, and a model for how a given observation sequence was generated by an appropriate HMM. A complete description of the model could be given as

$$\Lambda = (A, B, \pi).$$

(2.2)
CHAPTER 2. HIDDEN MARKOV MODELS

The observations \( O_i \) of the Hidden Markov Model as shown till now were discrete symbols from a finite alphabet \( v_1, v_2, \ldots, v_M \). For many problems, this assumption is not a straightforward one, and requires the quantization of continuous signals into discrete symbols. There exists, however, a very elegant modification to the basic HMM that enables the insertion of continuous signals directly into the model [61]. From this point on the symbols used in the model are continuous, from a known distribution. This distribution should obey some restrictions.

The most general representation of such a probability density function is:

\[
    b_i(O) = \sum_{m=1}^{M} c_{im} P[O, \mu_{im}, U_{jm}] ,
\]

where \( b_i \) is the probability of emitting an output vector (observation) \( O \) at state \( i \), \( c_{im} \) is the mixture coefficient for the \( m \)th mixture in state \( i \) and \( P \) is any log-concave or elliptically symmetric density, with mean vector \( \mu_{im} \) and covariance matrix \( U_{im} \) for the \( m \)th mixture component in state \( i \) [39].

2.3 Basic problems of the HMM

Use of the HMM usually raises a few questions about the model [60].

1. Given the observation sequence \( O = O_1 O_2 \cdots O_T \), and a model \( \Lambda = (A,B,\pi) \), how do we efficiently compute \( P(O|\Lambda) \), i.e. the probability of the observation sequence, given the model? This problem is referred to as the evaluation problem.

2. Given the observation sequence \( O = O_1 O_2 \cdots O_T \), and a model \( \Lambda = (A,B,\pi) \), how do we choose a corresponding state sequence \( Q = q_1 q_2 \cdots q_T \) which is optimal in some meaningful sense (i.e., best “explains” the observations)? This problem is referred to as the decoding problem.

3. How do we adjust the model parameters \( \Lambda = (A,B,\pi) \) to maximize \( P(O|\Lambda) \)? This problem is referred to as the learning problem, or the estimation problem.

The solution of these problems is important for the realization of the model. When several Hidden Markov models exist and an observation sequence is presented, the solution to the evaluation problem is used to determine the most probable model for describing the data. After finding the best model to fit the data, the solution to the decoding problem helps reveal the underlying structure of the problem, i.e. the hidden states of the model. The solution to the learning problem is used to build the models used in the previous problems.

2.4 Solution to the evaluation problem

The straightforward solution to that problem is:

\[
    P(O|\Lambda) = \sum_{allQ} P(O|Q,\Lambda)P(Q|\Lambda) = \sum_{q_1,q_2,\ldots,q_T} \pi_{q_1} b_{q_1}(O_1)a_{q_1,q_2} b_{q_2}(O_2)\cdots a_{q_{T-1},q_T} b_{q_T}(O_T) ,
\]
where \( q_1, q_2, \ldots, q_T \) is the permutation of all state sequences of length \( T \). The problem with this approach is the number of calculations needed, which is in the order of \( T N^T \). This may be infeasible even for small values of \( N \) and \( T \).

The Forward-Backward Procedure can be used to compute this problem much more efficiently. In this algorithm a forward variable \( \alpha_t(i) \) defined as:

\[
\alpha_t(i) \overset{def}{=} P(O_1 O_2 \cdots O_t, q_t = S_i | \Lambda),
\]

is computed. This variable can be used throughout the following procedure:

1. Initialization:

\[
\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N .
\]

2. Induction:

\[
\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \right] 1 \leq t \leq T - 1, \quad 1 < j < N .
\]

3. Termination:

\[
P(O|\Lambda) = \sum_{i=1}^{N} \alpha_T(i) .
\]

This procedure computes the required probability in its termination step and the calculation of this variable involves only an order of \( N^2T \) operations. This number is quite feasible for implementation. This procedure is a special case of dynamic programming [16, 66], which is a mathematical concept used for the analysis of sequential decision problems, based on the principle of optimality. 

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [17]. The use of this technique assures the correctness and optimality of the forward-backward procedure.

The computation of a backward variable can be carried out in a similar manner [60]. This variable is defined as:

\[
\beta_t(i) \overset{def}{=} P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \Lambda).
\]

Both forward and backward variables can be used to solve the first problem of the HMM.

### 2.5 Solution to the decoding problem

In order to solve the decoding problem, the notion of optimality has to be defined more specifically. Optimality may be defined in several ways, among them:

- The state sequence in which every state \( q_t \) is individually most likely.
- The state sequence that maximizes the expected number of correct pairs of states.
- The state sequence that maximizes the expected number of correct triplets of states.
The most widely used criterion for optimality is to find the single best state sequence (path), i.e., to maximize \( P(Q|O, \Lambda) \). The algorithm that solves this problem is called a Viterbi algorithm and it is a form of the previously described dynamic programming, for a stochastic system [71, 24].

The Viterbi algorithm finds the single best state sequence, \( Q = q_1, q_2, \ldots, q_T \), for the given observation sequence \( O = O_1, O_2, \ldots, O_T \), and uses the following quantity:

\[
\delta_t(i) \overset{\text{def}}{=} \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \Lambda],
\]

i.e., \( \delta_t(i) \) is the best score (highest probability) along a single path, at time \( t \), which accounts for the first \( t \) observations and ends in state \( S_i \). By induction we have:

\[
\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(O_{t+1}).
\]

To actually retrieve the state sequence, the track of the argument which maximizes 2.11 must be kept. This is done via the auxiliary array \( \psi_t(j) \).

The complete procedure is stated as follows:

1. Initialization:

\[
\begin{align*}
\delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\
\psi_1(i) &= 0.
\end{align*}
\]

Recursion:

\[
\begin{align*}
\delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T \quad 1 \leq j \leq N, \\
\psi_t(j) &= \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T \quad 1 \leq j \leq N.
\end{align*}
\]

2. Termination:

\[
\begin{align*}
P^* &= \max_{1 \leq i \leq N} [\delta_T(i)], \\
q^*_t &= \arg \max_{1 \leq i \leq N} [\delta_T(i)].
\end{align*}
\]

3. Path (state sequence) backtracking:

\[
q^*_t = \psi_{t+1}(q^*_{t+1}), \quad t = T - 1, T - 2, \ldots, 1.
\]

### 2.6 Analysis of the learning problem

This problem is involved in finding the model \( \Lambda \) that maximizes \( P(O|\Lambda) \) given a fixed \( O \), i.e., the likelihood function given the observation. This problem can be described, therefore, as finding the global maximum of the likelihood function (ML). There is no known way to analytically compute the global maximum of the likelihood function. It is possible, however, to find local maxima of this function, and two of these algorithms will be shown here. These algorithm are:

1. The Baum-Welch method, also known as the EM (expectation modification) method [21, 15, 73, 14].
2. The Segmental K-means technique [46, 40, 62].

The Baum-Welch algorithm provides an estimate which locally maximizes the likelihood function of the given observation sequence, and the segmental K-means algorithm provides an estimate which locally maximizes the joint likelihood of the observation sequence and the most likely state sequence.

### 2.6.1 The Baum-Welch algorithm solution

In order to build the Baum-Welch reestimation algorithm we first define $\xi_t(i, j)$, the probability of being in state $S_i$ at time $t$, and state $S_j$ at time $t + 1$, given the model and the observation sequence, i.e.

$$\xi_t(i, j) \overset{\text{def}}{=} P(q_t = S_i, q_{t+1} = S_j | O, \Lambda).$$  \hspace{1cm} (2.19)

This definition can be rewritten using the definitions of the forward and backward variables 2.5 and 2.9.

$$\xi_t(i, j) = \frac{a_t(i) a_{t+1} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \Lambda)} = \frac{\alpha_t(i) a_{t+1} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{t+1} b_j(O_{t+1}) \beta_{t+1}(j)}.$$  \hspace{1cm} (2.20)

The probability of being in state $S_i$ at time $t$, given the observation sequence and the model - $\gamma_t(i)$ is defined as

$$\gamma_t(i) \overset{\text{def}}{=} P(q_t = S_i | O, \Lambda) = \frac{\alpha_t(i) \beta_t(i)}{P(O | \Lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)},$$  \hspace{1cm} (2.21)

and can be related to $\xi_t(i, j)$ as:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j).$$  \hspace{1cm} (2.22)

The summation of $\gamma_t(i)$ over the time index $t$, gives the expected (over time) number of times that state $S_i$ is visited, or equivalently, the expected number of transitions made from state $S_i$. Similarly, summation of $\xi_t(i, j)$ over $t$ can be interpreted as the expected number of transitions from state $S_i$ to state $S_j$, that is:

$$\sum_{i=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i,$$  \hspace{1cm} (2.23)

and
\[
T-1 \sum_{i=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j. \tag{2.24}
\]

Using the above formulas (and the concept of counting event occurrences), the method for reestimation of the parameters of the HMM, i.e. a set of reasonable reestimation formulas for \( \pi, A \) and \( B \) is:

\[
\overline{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } t = 1 = \gamma_1(i), \tag{2.25}
\]

\[
\overline{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{2.26}
\]

and

\[
\overline{b}_{j(k)} = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } S_j} = \frac{\sum_{t=1}^{T} \gamma_t(j,v_k)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{2.27}
\]

If the current model is \( \Lambda = (A, B, \pi) \), then the reestimated model is \( \overline{\Lambda} = (\overline{A}, \overline{B}, \overline{\pi}) \), which is derived from Eq 2.25 - 2.27. It can be proven [15, 73] that either:

1. The initial model \( \Lambda \) defines a critical point of the likelihood function, in which case \( \overline{\Lambda} = \Lambda \).

2. Model \( \overline{\Lambda} \) is more likely than model \( \Lambda \) in the sense that \( P(O|\overline{\Lambda}) > P(O|\Lambda) \), i.e. the new model \( \overline{\Lambda} \) is more likely to produce the observation sequence.

Based on the above procedure, the probability of \( O \) being observed from the model can be improved until some limiting point is reached. The final result of this reestimation procedure is called a maximum likelihood estimate of the HMM.

In the implementation of continuous observation, the reestimation of the transition matrix \( A \) and the initial state distribution is identical to the one used for discrete observation Eq. 2.25-2.26. The reestimation of the continuous observation probabilities is as follows:

\[
\overline{a}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,v_k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}, \tag{2.28}
\]

\[
\overline{b}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,v_k)O_t}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{2.29}
\]

and
\[ \mathbf{U}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)(\mathbf{O}_t - \mu_{jk})P(O_t - \mu_{jk})}{\sum_{t=1}^{T} \gamma_t(j,k)} . \] (2.30)

Where \( \gamma_t(j,k) \) is the probability of being in state \( j \) at time \( t \) with the \( k \)th mixture component accounting for \( O_t \), i.e.,

\[ \gamma_t(j,k) \overset{\text{def}}{=} \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \left[ \frac{c_{jk} P(O_t, \mu_{jk}, \mathbf{U}_{jk})}{\sum_{m=1}^{M} c_{jm} P(O_t, \mu_{jm}, \mathbf{U}_{jm})} \right] . \] (2.31)

This term can be generalized to \( \gamma_t(j) \) in the case of a simple mixture, i.e. \( M = 1 \), and rewritten exactly as in Eq. 2.21. The second term in the definition of Eq. 2.31 is identically \( 1 \).

### 2.6.2 The segmental K-means algorithm

The segmental K-means [62] starts with a training set of observations (the same as is required for parameter estimation), and an initial estimate of all model parameters. Following model initialization, the set of training observation sequences is segmented into states, based on the current model \( \Lambda \). This segmentation is achieved by finding the optimum state sequence, via the Viterbi algorithm, and then backtracking along the optimal path. The observation vectors within each state \( S_j \) are clustered into a set of \( M \) clusters, where each cluster represents one of the \( M \) mixtures of the \( b_j(O_t) \) density. From the clustering, an updated set of model parameters is derived as follows:

- \( \hat{c}_{jm} \) - The number of vectors classified in cluster \( m \) of state \( j \) divided by the number of vectors in state \( j \).
- \( \hat{\mu}_{jm} \) - The sample mean of the vectors classified in cluster \( m \) of state \( j \).
- \( \hat{\Sigma}_{jm} \) - The sample covariance matrix of the vectors classified in cluster \( m \) of state \( j \).

Based on this state segmentation, updated estimates of the \( a_{ij} \) coefficients may be obtained by counting the number of transitions from state \( i \) to \( j \) and dividing it by the number of transitions from state \( i \) to any state (including itself). If the number of mixtures \( (M) \) is one, then the computation of these values is simplified.

An updated model \( \Lambda' \) is obtained from the new model parameters and the formal reestimation procedure is used to reestimate all model parameters. The resulting model is then compared to the previous model (by computing a distance score that reflects the statistical similarity of the HMMs, based on the probability functions of the parameters). If the model distance score exceeds a threshold, then the old model \( \Lambda \) is replaced by the new (reestimated) model \( \Lambda' \), and the overall training loop is repeated. If the model distance score falls below the threshold, then model convergence is assumed and the final model parameters are saved. The convergence of this algorithm is also assured [48] as is in the Baum-Welch algorithm.

### 2.7 Clustering techniques used in HMM

Both the Baum and the segmental K-means algorithms are initialized with an initial model \( \Lambda_{\text{start}} \). This initialization is crucial to the results of the algorithms due to the locality problems of the above
CHAPTER 2. HIDDEN MARKOV MODELS

mentioned algorithm. The computation of the initial model is usually carried out by a clustering algorithm. Clustering of data is the process of partitioning a given set of data samples into several groups [22, 38, 12, 47]. Each group usually indicates the presence of a distinct category in the data. One of the common clustering algorithms is defined as follows:

1. set \( n = 1 \)

2. Choose some initial values for the model parameters - \( \mathbf{\Lambda}_1^{(n)}, \mathbf{\Lambda}_2^{(n)}, \ldots, \mathbf{\Lambda}_K^{(n)} \), where \( K \) is the number of clusters computed. Each of these \( \mathbf{\Lambda}_j \) represents one class of the algorithm, and includes the parameters characterizing that class.

3. Loop: classify all the samples, by assigning them to the class of the closest divergence. This classification defines an error measure, which is the total divergence of all the samples from their matched classes.

4. Recompute the model parameters as the average of all the samples in their class. The new set of parameters is \( \mathbf{\Lambda}_1^{(n+1)}, \mathbf{\Lambda}_2^{(n+1)}, \ldots, \mathbf{\Lambda}_K^{(n+1)} \)

5. If the distance between the error measures of sets \( \mathbf{\Lambda}_j^{(n+1)} \) and \( \mathbf{\Lambda}_j^{(n)} \) is below some threshold, stop, and the result is \( \mathbf{\Lambda}_j^{(n)} \).

6. \( n \rightarrow n + 1 \)

7. go to beginning of Loop (step 3).

This algorithm is sometimes referred to as the K-means algorithm [22, 49], where \( K \) stands for the number of clusters sought. Other names for this algorithm are the LBG [47], the ISODATA algorithm [22] or VQ (Vector Quantization) algorithm. The algorithm described above is referred to as hard clustering, due to the fact that each data point is related to one cluster. There exists, however, a different clustering method referred to as soft clustering, in which each data point is related to all of the centroids, each with a certain probability. In this work the soft clustering technique was used.

2.7.1 KL-divergence

The process of clustering data samples necessitates a divergence measure, which is used when classifying each data sample to the matching cluster (step No. 3 in the K-means algorithm). An optimal measure for the case of known observation probabilities is shown below.

For a given multidimensional process the probabilities are \( q_1, q_2, \ldots, q_K \), where \( q_i \) may form any distribution function. The observation of this process is \( n_1, n_2, \ldots, n_K \) from any symbol space. In order to classify the observation vector, its probability

\[
P(n_1, n_2, \ldots, n_K | q_1, q_2, \ldots, q_K),
\]

is computed. Regarding this observation as the output of \( N \) Bernoulli trials, where

\[
N = \sum_{i=1}^{K} n_i,
\]
CHAPTER 2. HIDDEN MARKOV MODELS

the probability measure of Eq. 2.32 is:

\[ P(n_1, n_2, \ldots, n_K, q_1, q_2, \ldots, q_K) = \frac{N!}{\prod_{i=1}^{K} n_i!} \prod_{i=1}^{K} q_i^{n_i}. \]  

(2.34)

Taking the natural logarithm of both sides, and using Stirling's approximation, i.e.

\[ N! \approx \sqrt{2\pi N} \left( \frac{N}{e} \right)^N, \]  

we get:

\[ \log P(\pi, q) = N \log(N) - N - \sum_i n_i \log(n_i) + \sum_i n_i \log(q_i) + o(N) = \]  

(2.36)

(due to Eq. 2.33)

\[ = N \log(N) - \sum_i n_i \log(n_i) + \sum_i n_i \log(q_i) + o(N). \]  

(2.37)

Defining a new set of observations \( p_i \) as:

\[ p_i \overset{df}{=} \frac{n_i}{N}, \]  

(2.38)

Eq. 2.37 can be rewritten as:

\[ -\sum_i n_i \log \left( \frac{n_i}{N} \right) + \sum_i n_i \log(q_i) + o(N) = -N \left[ \sum_i p_i \log \left( \frac{p_i}{q_i} \right) \right] + o(N). \]  

(2.39)

The probability measure of Eq. 2.32 can be written as:

\[ P(\pi|q) \approx e^{-ND[p||q]}, \]  

(2.40)

where \( D[p||q] \) is defined as:

\[ D[p||q] \overset{df}{=} \sum_{A\in\mathcal{X}} p(x) \log \left( \frac{p(x)}{q(x)} \right). \]  

(2.41)

This expression is called the Kullback-Liebler divergence measure (KL) [45]. It is also sometimes referred to as the cross-entropy of the distributions [48].

A different way of reaching the same divergence measure is by using Bayes rule. Suppose that two hypotheses exist concerning the distribution of data:

1. \( H_1 \): The data are distributed according to \( P(x^n|H_1) \).
2. \( H_2 \): The data are distributed according to \( P(x^n|H_2) \).
In the context of clustering problems these hypotheses are the different centroids of the clusters, and the problem is to classify a given observation \( x \) to one of the centroids.

Given \( n \) independent samples from one of the distributions, the task is to evaluate the term \( P(H_1|x^{(n)}) \) which is, according to Bayes’ rule:

\[
P(H_1|x^{(n)}) = \frac{P(x^{(n)}|H_1)P(H_1)}{\sum_{i=1}^{2} P(H_i)P(x^{(n)}|H_i)},
\]

because the samples are independent, the probability can be turned into a multiplication:

\[
P(x^{(n)}|H_i) = \prod_{j=1}^{n} P(x_j|H_i).
\]

Thus the term in 2.42 can be rewritten as:

\[
P(H_1|x^{(n)}) = \frac{P(H_1) \prod_{j=1}^{n} P(x_j|H_1)}{\sum_{i=1}^{2} P(H_i) \prod_{j=1}^{n} P(x_j|H_i)}.
\]

By dividing the nominator and denominator by the same expression we get:

\[
P(H_1|x^{(n)}) = \frac{1}{1 + \frac{P(H_2)}{P(H_1)} \prod_{j=1}^{n} \frac{P(x_j|H_2)}{P(x_j|H_1)}} = \frac{1}{1 + e^{\sum_{j=1}^{n} log \left( \frac{P(x_j|H_2)}{P(x_j|H_1)} \right)}}.
\]

Due to the fact that \( P(H_1) \) and \( P(H_2) \) do not depend upon the data samples, the logarithm of their portion could be written as a constant \( \theta \). Thus, 2.47 can be rewritten as:

\[
\frac{1}{1 + e^{\sum_{j=1}^{n} log \left( \frac{P(x_j|H_1)}{P(x_j|H_2)} \right)}} - \theta = \frac{1}{1 + e^{-X}} - \theta,
\]

where

\[
X = \sum_{j=1}^{n} log \left( \frac{P(x_j|H_1)}{P(x_j|H_2)} \right) + \theta.
\]

According to the strong law of large numbers, the sample average converges to the expected value with respect to the distribution which the data were taken from (assuming that the data were taken from \( H_1 \)), thus indicating that as the summation factor grows this expression becomes:

\[
X = n \sum_{\text{All } x} p(x|H_1) \log \left( \frac{P(x|H_1)}{P(x|H_2)} \right) - \theta.
\]
Thus the classification of data points to their clusters is governed by this term, or more generally by the term in Eq. 2.41.

The KL divergence optimality can be seen in yet another way, based on information theory [19]. The entropy of a distribution \( p(x) \):

\[
E[p] = - \sum_x p(x) \log p(x) ,
\]

(2.51)

can be thought of as the number of bits needed to code it, or describe it. The KL divergence can be rewritten as:

\[
D[p||q] = \sum_x p(x) \log \left( \frac{p(x)}{q(x)} \right) = \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) ,
\]

(2.52)

that is, the KL divergence can be considered as the penalty paid for trying to code (describe) \( p(x) \) using the optimal code of \( q(x) \). This penalty is the difference between the length of coding \( p(x) \) using \( q(x) \) (non-optimal), and the length of coding \( p(x) \) using the correct optimal coding.

2.7.2 Characteristics of the KL divergence

The KL divergence is always a non-negative function, and it is equal to zero if and only if the two probability measures \( p \) and \( q \), are identical, but it is not necessarily symmetric, i.e. \( D[p||q] \neq D[q||p] \) [19]. The last fact prevents the usage of KL divergence as a real distance measure, although it was shown to be optimal in several ways.

This limitation is due to the fact that the KL divergence is optimal in the following manner: the KL divergence indicates how likely it is to receive samples distributed according to \( p \) where the real probability is \( q \). This question is clearly not a symmetric one, and therefore the optimal answer is not always a symmetric one.

It would, however, be of interest to observe an example in which the KL divergence is a symmetric measure, and its relevance to the clustering of data. In this example the KL divergence of two different Gaussian distributions with the same variance is computed.

Let \( p \) and \( q \) be two such distributions, i.e.

\[
p(x) = \frac{1}{\sqrt{2\pi\sigma_1}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} ,
\]

(2.53)

and

\[
q(x) = \frac{1}{\sqrt{2\pi\sigma_2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} .
\]

(2.54)

In the special case where \( \sigma_1 = \sigma_2 = 1 \), the KL divergence is:

\[
D[p||q] = \sum_{\text{All } x} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{2}} \log \left( \frac{\sqrt{2\pi} e^{-\frac{(x-\mu_1)^2}{2}}}{\sqrt{2\pi} e^{-\frac{(x-\mu_2)^2}{2}}} \right) =
\]
The KL divergence in this example is the Euclidean distance measure. That is, in the case of two normal distributions, the optimal divergence between them is the commonly used distance measure.

2.8 Hidden Markov Model of multivariate Poisson probability observation

The notion of Poisson probability is introduced here by means of completeness of description of data modeling. One of the common assumptions regarding the probability distribution of spike trains is the Poisson distribution:

\[ P_\lambda(n) = \frac{e^{-\lambda} \lambda^n}{n!} . \]  

That is, the probability of finding a certain number of spikes \( n \) in some interval is governed by this probability function with a mean of \( \lambda \). Throughout this section, \( \lambda \) will be referred to as the firing rate of cells.

Under the Poissonian distribution the clustering divergence measure will be shown, and a simple ML estimation of parameters will be reached.

2.8.1 Clustering data of multivariate Poisson probability

The divergence was computed on the basis of the assumption that the firing rate followed an uncorrelated multivariate Poisson process:

\[ P(n_1, n_2, \ldots, n_d \mid \lambda) = \prod_{i=1}^{d} \frac{e^{-\lambda_i} \lambda_i^{n_i}}{n_i!} , \]  

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d) \) and \( d \) is the number of uncorrelated Poisson processes. For 1-dimensional Poisson distribution \( (d = 1) \), the divergence is given by:

\[ D[p||q] = \sum_x p(x) \log \frac{p(x)}{q(x)} = \sum_x \frac{e^{-\lambda_1} \lambda_1^x}{x!} \log \frac{e^{-\lambda_2} \lambda_2^x}{x!} = \lambda_2 - \lambda_1 + \lambda_1 \log \frac{\lambda_2}{\lambda_1} , \]  

where

\[ p(x) = \frac{e^{-\lambda_1} \lambda_1^x}{x!} , \]  

and

\[ q(x) = \frac{e^{-\lambda_2} \lambda_2^x}{x!} . \]

The uncorrelated multivariate divergence is simply the sum of divergences for all the 1-dimensional probabilities.
CHAPTER 2. HIDDEN MARKOV MODELS

This multivariate Poisson divergence is not a symmetric measure, i.e. $D[p||q] \neq D[q||p]$. Thus in Eq. 2.58, $\lambda_1$ is the firing rate of the sample vector and $\lambda_2$ is the firing rate of the centroid of the cluster checked. The KL divergence measures the probability of assigning a firing rate vector (from the data) to a specific centroid.

2.8.2 Maximum likelihood of multivariate Poisson process

A derivation of ML estimate of multivariate Poisson process with a pair-wise cross-correlation will be shown here. The probability of an observation vector $\mathbf{n}$ is defined as:

$$P(\mathbf{n} | \lambda, \tilde{\alpha}) \overset{def}{=} \prod_{i=1}^{N} \frac{e^{-\lambda_i \lambda_i n_i}}{n_i!} \exp\left[- \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} n_i n_j \right] \frac{1}{Z},$$  \hfill (2.61)

where $\lambda_i$ is the vector of firing rates of the multivariate Poisson distribution (as described in the previous subsection), $\alpha_{ij}$ is the cross-correlation matrix indicating the correlation between the process $i$ and the process $j$, and $Z$ is defined as:

$$Z(\lambda, \tilde{\alpha}) \overset{def}{=} \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_N=1}^{M_N} P(\mathbf{n} | \lambda, \tilde{\alpha}).$$  \hfill (2.62)

$M_i$ is defined as the minimal value which satisfies

$$\sum_{j=1}^{M_i} \frac{e^{-\lambda_i \lambda_j}}{j!} < \epsilon \quad \forall \lambda_i \in \lambda.$$  \hfill (2.63)

The log likelihood function of this process is

$$\log L(\tilde{\pi} | T) = \sum_{i=1}^{T} \log P(\mathbf{n} | \lambda, \tilde{\alpha}) = \sum_{i=1}^{T} \left( (\sum_{i=1}^{N} \log P_i(n_i)) - (\sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} \alpha_{ij} n_i n_j) - T \log(Z) \right).$$  \hfill (2.64)

The ML is computed by satisfying the terms

$$\frac{\partial \log L}{\partial \lambda_i} = 0, \quad \frac{\partial \log L}{\partial \alpha_{ij}} = 0.$$  \hfill (2.65)

This ML computation results in

$$\frac{\partial \log L}{\partial \lambda_i} = \left( \sum_{i=1}^{N} \frac{n_i}{\lambda_i} \right) - T - \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_N=1}^{M_N} [\prod_{j=1, j \neq i}^{N} \frac{e^{-\lambda_j \lambda_j n_j}}{n_j!}]$$

$$- \sum_{j=1, i \neq j}^{N} \left( \exp[\sum_{j=1}^{N} \sum_{k=1, j \neq k}^{N} a_{jk} n_j n_k] \left( \frac{1}{n_i!} e^{-\lambda_i \lambda_i n_i} \right) \right)$$

$$= 0,$$  \hfill (2.66)

and
\[
\frac{\partial \log L}{\partial \alpha_{ij}} = \left( \sum_{t=1}^{T} - n_i^n t_j^n - \frac{T}{Z} \sum_{n_1=1}^{M_1} \sum_{n_2=1}^{M_2} \cdots \sum_{n_N=1}^{M_N} \left( \prod_{k=1}^{N} \frac{e^{-\lambda_k \lambda_k^k}}{\gamma_k^k} \right) \right) \\
= \left( \exp \left[ \sum_{k=1}^{N} \sum_{l=1}^{N} \alpha_{kl} n_k n_l \right] \right) \left( -n_i^n t_j^n \exp \left[ -\alpha_{ij} n_i^n t_j^n \right] \right) = 0.
\]

Eq. 2.66.2.67 can be solved using any gradient descent, such as conjugate gradient methods, see [59, pages 254–259].
Part II

Model Implementation
Chapter 3

Physiological Data and Preliminary Analysis

3.1 Frontal cortex as a source of data

The aim of the research presented here, was to apply HMM models to physiological data in order to be able to predict cognitive states of the organism. One of the valuable sources of information about the activities of the brain is the frontal cortex (FC), which is located in the anterior third of the brain. The FC is known to contain high association areas, i.e. areas which are not directly connected to sensory or motor systems, but receive connections from, and send connections to, many other cortical areas. The FC has been divided into three broad regions: the motor cortex (MC), the premotor cortex (PMC) and the prefrontal cortex (PFC). Two of these areas were investigated in this work, the PMC and PFC [33, 56]. Fuster claims that "the PFC is critical for temporal organization of behavior. It mediates cross-temporal sensorimotor contingencies, integrating motor action (including speech in humans) with recent sensory information" [26, 27, 28]. This area is also involved in the composition and execution of plans [18]. The PMC is essential for the preparation of movement and the ability to develop an appropriate strategy for movement [33]. These areas are characterized by multiple connections with many other cortical areas, most of which are associative areas.

Information about neural activity in the FC is gained through electrophysiological recording of cells in a behaving animal (a conscious animal engaged in a specific pre-defined task) [26, 70]. This enables us to compare the recordings of cells to the actual activity of the animal. A pre-defined task that is sometimes used when recording from the PMC and the PFC is the spatial delayed response task [52, 53, 51, 25]. This task consists of the following stages:

1. Initialization of task.
2. A sensory cue.
3. Delay period.
4. An action cue.
6. Reward in the case where the response was correct.

It is known that both areas under investigation (PMC and PFC), play an important role in the performance of this task.

### 3.2 Description of physiological experiment

The data used in the present analysis were collected in the Higher brain function laboratory at the Hadassah Medical School. The present chapter describes the experiments carried out there, and the following chapter will discuss the application of the model to these data. Recordings were taken from two Rhesus monkeys (*Macaca Mulatta*), trained to perform a spatial delayed-response task. The experiments were conducted by Y. Prut, I. Halman and H. Slovin as part of their Ph.D work under the supervision of M. Abeles, H. Bergman and E. Vaadia.

#### 3.2.1 Behavioral modes of the monkey

The monkeys were trained to perform a spatial task, in which they had to alternate between two behavioral modes. The monkey initiated the trial by pressing a central key, and a fixation light was turned on in front of it. Then, after 3-6 seconds, a visual stimulus appeared, in the form of a cue light, either on the left or on the right. The stimulus was presented for 100 ms. The stimulus was followed by a delay period of 1-32 seconds, in factors of 2 (i.e. 1, 2, 4, 8, 16, 32 seconds). Following the delay period, the fixation light was dimmed, and the monkey was required to touch the visual cue light (“Go” mode), or keep its hand on the central key regardless of the external stimulus (“No-Go” mode). The monkey was rewarded for the correct behavior with a drop of juice. After 4 correct trials, all the lights in front of the monkey blinked, signaling to the monkey to change its behavioral mode - so that if it had started in the “Go” mode it now had to switch to “No-Go” mode, and vice versa. A diagram of the behavioral modes is presented in figure 3.1.

The training period for each monkey lasted 2-4 months. At the end of the training, the monkeys were able to perform this complicated task with low error rates (around 10%).

#### 3.2.2 Data from recording session

At the end of the training period, the monkeys were prepared for electrophysiological recording. Surgical anesthesia was induced. A recording chamber was fixed above a hole drilled in the skull to allow for a vertical approach to the frontal cortex. After recovery from surgery, recording sessions began, and the activity of the cortex was recorded while the monkey was performing the previously-learned routine. Each recording session was held on a separate day, and typically lasted 2-4 hours. During each of the recording sessions, six microelectrodes were used simultaneously. The electrodes were placed in a circle with a radius of 0.5 mm. With the aid of two pattern detectors and four window-inscriminates, the activity of up to 11 single cells (neurons) was concomitantly recorded [68, 4]. This activity consists of the discrete times of action potentials of the cells, referred to as the *spike train* of the cell. The recorded data includes the following information:

1. The spike trains of all the cells.
2. The recording session events. These events are either the stimuli presented to the monkey or the responses of the monkey.
CHAPTER 3. PHYSIOLOGICAL DATA AND PRELIMINARY ANALYSIS

3. The electro-occulogram (EOG) which indicates the direction of the eyes’ gaze in the horizontal plane.

The data analysis presented here was carried out after the experiments had ended. It is important to note that the recordings do not reflect the learning process, but rather the cortical activity of the well-trained monkey while performing its task.

3.3 Basic tools of analysis

From the variety of analysis tools used to analyze the data, the dot-display tool and the cross-correlation tool formed the basis for the work described in this paper. The theory and implementation of these tools is described briefly in the following section.

3.3.1 Dot display tool

The basic tool used for presentation and analysis of the spike trains of several cells is the dot-display. This tool enables the user to interactively display spike trains of several cells according to a specified condition list which defines time slices from the full spike train.

Some of the major conditions used in selection of data are:

- The sequence of events that define the time section displayed. One of these events is defined as a center event.
- The time interval before and after the center event.
- The behavioral mode of the monkey at the time of recording.

An example of a dot-display of a single cell is presented in figure 3.2, and includes the following information:
• Cell index.

• Spike trains of the specified cell, drawn one on top of the other. Each one of the spikes in the spike train is presented as a small vertical line.

• The number of spike trains drawn - in this example there are 233 spike trains.

• The sequence of events defining the selection criteria for the spike trains.

• The recording session data are divided into separate files, and the list of files used for this display is shown.

• The time scale is shown on the abscissa. This example presents spike trains from 4000 ms before the center event, until 5000 ms after that event.

The peri-stimulus-time (PST) histogram is drawn above the spike trains. In this example, a bin width of 50 ms was used, i.e. in the 9000 ms presented in this figure, 180 such bins exist. The computation process involves counting the spikes in each such bin and dividing the result by the number of repetitions for each bin. The result is transformed into units of spikes-per-second by dividing it by the time duration of the bins. The histogram is automatically scaled according to the maximum firing rate value, which is given on the upper left side of the histogram, in units of spikes-per-second. In this example, the maximum firing rate is 4 spikes-per-second.

3.3.2 Temporal cross correlation tool

The temporal cross-correlation function between cells measures the correlation between spike trains of cells (discrete events), and can be computed in bins of variable length. The figures representing the cross-correlation function with a specific bin width will henceforth be called a cross-correlograms [69].

The temporal cross-correlogram between two cells \( i \) and \( j \) at time bin \( i \) is defined as follows:

\[
R_{ij}(l) = \frac{1}{N_i \Delta t} \sum_{t=1}^{N_i} S'_i(t) S''_j(t+l) .
\]  

(3.1)

Where \( S'_i(t) \) is the event that cell \( i \) emitted a spike in the time bin \((t, t + 0.001)\), and \( S''_j(t + l) \) is the event that cell \( j \) emitted a spike in the time bin \((t + i \Delta t, t + (i + 1) \Delta t)\). The absolute time mentioned above is in seconds, i.e. 0.001 is 1 ms. \( N_i \) is the number of spikes of cell \( i \) (that are used as “triggers” for the averaging) and \( \Delta t \) is the time bin width. Where \( l \Delta t \) extends from \(-T\) to \(+T\), the cross-correlation functions for positive \( l \Delta t \) values \((0 < l \Delta t < T)\) express the average firing rate of cell \( j \) at time \( l \Delta t \), given that cell \( i \) fired a spike at time 0. Similarly, the cross-correlation function at \((0 < l \Delta t < 0)\) gives the firing rate of cell \( i \), given that cell \( j \) fired a spike at time 0.

The shape of the cross-correlogram provides indications for the functional connectivity between neurons. Typical shapes of cross-correlograms feature peaks or troughs near the origin. When the peaks or troughs extend to both sides of the origin, they are usually interpreted as reflecting a shared input to the two neurons. Narrow peaks or troughs, restricted to one side of the cross-correlogram, are usually interpreted as reflecting a direct synaptic interaction.

An example of a cross-correlogram is presented in figure 3.3, and includes the following information:
Figure 3.2: An example of a dot-display.

- The indexes of the cells used to compute the correlogram.
- The average firing rate of each cell.
- The time scale of the cross-correlogram is shown on the abscissa. In this example, from -500 ms to +500 ms.
- The firing rate is shown on the ordinate in units of spikes-per-second.
- The total duration of spike trains used to compute this correlogram.
- Bin size of Gaussian smoothing applied to the counting of the correlation function.
- An average line which is the average of the second cell.
Figure 3.3: An example of a cross-correlation function (correlogram) of two cells.

- Significance lines with a confidence level of 0.005 for each side of the average, computed on the tails of the correlation function, i.e. from -500 to -400 ms, and from +400 to +500 ms.

3.4 Quality of cells recorded

The quality of the recording is parameterized in several different non-overlapping ways. These parameters are:

1. The isolation score given to a cell during a recording session.
2. The stationarity of the cell as measured afterwards on the dot-display of the cell.
3. The firing rate of the cell.
4. The tendency of the cell to change its firing rate in response to events.

These parameters were integrated into a total score.

3.4.1 Isolation score given to cells during recording session

The first parameter is the isolation score that each cell is given during the recording session. This score is evaluated by the experimenter based on the percentage of spikes falling within a threshold
of probability, inside the window discriminator. The scores are:

A : Very well isolated single neuron with a large, unmistakable spike.

B : Most of the time a well isolated single neuron can be identified, but occasionally a spike from another cell may intrude.

B- : The same as B, but some of the spikes are lost due to thresholds in the spike sorters.

C+ : Two neurons, but most of the spikes originate from one neuron.

C : Two neurons which cannot be observed separately.

C- : Most of the time two neurons are counted, but occasionally other neurons are counted as well.

D : More than two neurons together.

E : Many small neurons - with much background noise.

Cells of types A and B are sure to be single, well isolated neurons, varying only in the shape of their spikes. The other types should be used, therefore, with caution.

Cells are given isolation scores several times during each recording session, due to the fact that these scores may vary over time. For a cell to be of use, it is usually required to remain within a small variance of score values. While recording the activity of the cells there are other types of quality indications:

- The reliability of the discrimination between cells.
- Which cells tend to contaminate one another.
- The pattern of the spike activity.

This information is also taken into account when the cells are chosen, but in a more qualitative way. For example two cells of type C were not used, if they were recorded from the same electrode, and the experimenter noted that they tended to contaminate one another, because this contamination may cause a severe artifact in their cross-correlation function.

### 3.4.2 Stationarity of a cell

Another measure of the acceptability of cells for analysis, is their stationarity during the recording session. This measure is extracted from the data after the recording session is over. Nonstationarity may be due to a non-stable recording - a recording of a cell that was injured, for example. An injury to a cell may have occurred while the electrodes were penetrating the tissue, and so the spike train of this cell would not be reliable. Instability may also result from a small movement in the tissue recorded. Another possible reason for nonstationarity may be an actual nonstationarity in the cell activity.

An example of nonstationarity may be seen in figure 3.4. In this figure, a cell that was given an isolation score of B is shown. Although this is a high isolation score, it can be seen clearly that the cell changes its firing rate dramatically during the recording session. This dramatic change precludes the use of all the cell spike trains as a basis for further analysis. It is possible to use
the spike trains of this cell only after they are broken down into several intervals. These intervals are shown in the figure as markers near the ordinate. There are three separate intervals on this dot-display. The first and third intervals may be used together, or each of these intervals may be used separately.

Figure 3.4: An example of nonstationarity in the firing rate of a cell, in which three distinct sections (marked in the left side of the dot-display) can be seen.

### 3.4.3 Firing rate of cells as a quality measure

The firing rate of a cell is another parameter to be taken into consideration when choosing the cells to be modeled. The firing rate should be as high as possible, to enable further processing, either sustained during the whole recording session, or raised as a response to some event. The time constants of the phenomena sought by the model are in the range of several hundreds of ms - so that a cell that fires in rates lower than several spikes-per-second, holds no useful information about these changes.

An example of different firing rates of two cells from the same recording session is presented in figure 3.5. The upper cell (No. 4) shows a typical firing rate of 10 spikes-per-second, and a maximum rate of 30 spikes-per-second, which is acceptable for the model. The lower cell (No. 3),
however, is firing at a much lower rate - its maximum rate is 0.5 spikes-per-second. This low firing rate, combined with a lack of specific responses, precludes the use of this cell in the model.

![Graph showing firing rates of two cells](image)

Figure 3.5: An example of different firing rates of two cells from the same recording session.

### 3.4.4 Responsiveness of cells to events

Modulation of firing rates around objectively measurable events is one of the important factors taken into consideration when scoring a cell. The significance of responsiveness depends on the model of information-processing in the cortex. If the model puts the emphasis on the firing rates of the cells, then the change in the firing rate would be considered evidence for response to events. If the model emphasizes exact temporal structures of a series of spikes, than this temporal structure would be considered a key to the responsiveness of cells. The present research uses the first model as a basis for measuring responsiveness.

Figure 3.6 shows the activity of two cells whose spike trains were gathered from the same recording session, near the same external cue. The lower cell (No. 5) shows a prominent response to the external stimulus. The external stimulus in this example was the cue given to the animal (either from the right or from the left), as part of the behavioral mode described in figure 3.1.
The cue of every trial is used here as time 0. The changes in the firing rate of this cell match the different delay times (1, 2, 4, 8 seconds) after the cue, at which the "Go" signal was presented. Furthermore, it can be seen in the dot-display that the response is characterized by a steep rise and a slow decay. The upper cell (No. 15) on the other hand, shows only a very weak response to the stimulus.

Figure 3.6: An example of two cells that show different responses to the same stimulus.
Chapter 4

Application of the Model

4.1 Data selection stage

The data used for modeling came from the recording sessions of 2 monkeys. The behavioral paradigm performed by the monkey in all the recording sessions was the *Delayed Release Paradigm* presented in figure 3.1 (Chapter 3 Section 2). The data used in the modeling included only the spike trains of all the cells recorded throughout each recording session. A quality check was carried out for each recording session separately, according to the following stages:

1. Only cells that were given isolation scores equal to or higher than C in the recording session (see Chapter 3 Section 4) were used.

2. The isolation score of the cell had to be stable over time. If that score was not stable during the whole session, stable intervals of the session were chosen.

3. Only stationary cells were used (see Chapter 3 Section 4). The stationarity check was performed using the dot-display tool. If the cells were not stationary during the whole session, only stable intervals of the session were chosen.

4. Cells recorded from the same electrode were checked for contamination evidence (see Chapter 3 Section 4), if any existed, at least one of these cells was not used.

5. The firing rate of the cell had to be higher than several spikes-per-second. The threshold used was 1 spike-per-second. It was not necessary that the average firing rate exceed the threshold, and is was considered sufficient if the firing rate of the cell exceeded the threshold only in a few sections of the spike trains, e.g., near some external stimulus, averaged throughout the whole recording session.

This quality check was applied to 8 recording sessions, out of which 6 were found to include *sufficient data* for further processing. *Sufficient data* in this context was defined as follows:

1. More than 5 well isolated cells.

2. The intervals chosen held more than 5000 *seconds* of stable isolation scores and stationary recording.
Following the quality check, the data were divided into trials made in the “Go” mode and the “No-Go” mode. From this point onwards, the spike trains of these behavioral modes were processed separately. The result of this selection can be seen in table 4.1.

<table>
<thead>
<tr>
<th>Session</th>
<th>Cells</th>
<th>States</th>
<th>“Go” Length</th>
<th>“No-Go” Length</th>
<th>Total Record Length</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>8</td>
<td>6</td>
<td>3907</td>
<td>4097</td>
<td>8004</td>
<td>PMC</td>
</tr>
<tr>
<td>bl25</td>
<td>8</td>
<td>7</td>
<td>6202</td>
<td>5131</td>
<td>11333</td>
<td>PMC</td>
</tr>
<tr>
<td>bl29</td>
<td>8</td>
<td>8</td>
<td>2557</td>
<td>2508</td>
<td>5065</td>
<td>PMC-MC</td>
</tr>
<tr>
<td>bl39</td>
<td>6</td>
<td>7</td>
<td>5987</td>
<td>5864</td>
<td>11854</td>
<td>PMC</td>
</tr>
<tr>
<td>cl77</td>
<td>6</td>
<td>6</td>
<td>9946</td>
<td>6566</td>
<td>16512</td>
<td>PFC</td>
</tr>
<tr>
<td>cl85</td>
<td>8</td>
<td>6</td>
<td>3908</td>
<td>4654</td>
<td>8562</td>
<td>PFC</td>
</tr>
<tr>
<td>Total</td>
<td>44</td>
<td>40</td>
<td>32507</td>
<td>28820</td>
<td>61327</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Presentation of the recording sessions used throughout the experiments. For each recording session, the number of well-isolated cells used is indicated, as is the total length of the session (in seconds) and the cortical area from which the data were recorded. These time tables are presented separately for the “Go” mode and the “No-Go” mode.

It is important to note that the cells were not chosen according to any criterion of responsiveness - some of the cells chosen were responsive to an external stimulus and some showed only weak responses to the stimulus.

### 4.2 Computation of parameters used in the model

The model assumes that the spike trains are distributed according to a Poisson distribution [1]. The distances between spikes are distributed exponentially and their number in each frame, $n$, depends only on the mean firing rate $\lambda$, through the distribution

$$P_\lambda(n) = \frac{e^{-\lambda} \lambda^n}{n!}. \quad (4.1)$$

The short-time firing rate of the cell, which is the basic parameter of the multivariate Poisson process, was computed. The computation was done with a sliding window of 500 ms width, and with transition of 100 ms. This computation transformed the day’s firing activity into a series of vectors, each $N$ dimensional - where $N$ is the number of cells in the experiment. An example of parameter computation is presented in figure 4.1. This figure presents a dot-display of 8 cells, recorded simultaneously. For each of these spike trains, the short-time firing rate of the spike train is plotted as a dashed line.

In this example, a time scale of 4 seconds was used. This computation therefore produces 40 vectors, each of them 8 dimensional. These vectors were the point-wise measurements for the next stages of computation.
4.3 Clustering of data vectors

The K-means algorithm (described in Chapter 2 Section 7) was used to cluster the data. The initialization of the algorithm, i.e. choosing the first set of model parameters, was done by randomly choosing $K$ vectors from the vectors’ space. The sets of model parameters will be referred to as the centroids of the model.

The K-means algorithm attempts to minimize the total divergence measure function, but unfortunately can reach only a local minima of this function (Chapter 2 Section 7). This fact makes the algorithm highly dependent upon the starting condition, i.e. the centroids of the first iteration $\bar{\lambda}_1^{(1)}, \bar{\lambda}_2^{(1)}, \ldots, \bar{\lambda}_K^{(1)}$, chosen at stage 2 of the K-means algorithm. In order to overcome this problem, the execution of the algorithm was repeated several times, starting from different initial centroid values. The number of repetitions was dependent on the convergence level of the divergence measure function and the centroids reached. Thus, if the same divergence measure and centroids were reached, the algorithm converged. The algorithm was first repeated 5-6 times, the halting condition was:

1. Each iteration produced a measurable result (the average divergence of observation vectors to their centroids).
2. The results were ordered from lower to higher.
3. If the range of the lower 60% of the scale was below a threshold, the process was halted. The
CHAPTER 4. APPLICATION OF THE MODEL

threshold used was 0.5.

4. The algorithm was repeated 5 more times and all the results were checked beginning with step No. 1.

The number of clusters \( (K) \) used in the algorithm is fixed throughout one repetition. In order to find the optimal number, several values were tested. For each of these values the algorithm was repeated several times, in order to reach convergence (as described above). The optimal number of clusters was calculated based on the error measure functions for the different values tested, taking into consideration the number of clusters. As the number of clusters increases, the divergence measure is reduced, but it is necessary to find some balance between the two. This balance was computed based on the reduction rate: when the number of states is small this reduction is usually logarithmic, and when this number increases, this reduction rate becomes linear or less. The balance was taken as the point where the change between logarithmic reduction rate and linear reduction rate occurs. An example of that process is shown in figure 4.2. In this figure two sections can be observed:

1. Number of states from 1 to 8, where the reduction is almost logarithmic.

2. Number of states from 9 to 15, where the reduction is almost linear.

The number of states used in this example was therefore 8. In order to classify each sample to a specific centroid, a specific divergence measure is needed. This divergence was computed in Chapter 2 Section 8, Eq. 2.58.

An example of the execution of the clustering algorithm is shown in figure 4.3. In this figure, two executions of the K-means algorithm are shown. For each of them the error measure is shown as a function of the iteration of the algorithm. The data used for the two executions came from different behavioral modes of the same recording session. The behavior shown in the graph is typical of this kind of algorithm: the first few iterations reduce the divergence measure function dramatically and the last iterations are the “fine tuning”.

This is a special case of a method called distributional clustering, recently developed in a more general setup [67].

4.4 Implementation of a full HMM

The next step in this framework was to compute a full Hidden Markov Model, consisting of the hidden states themselves and of the transition probabilities between them. In order to compute this model, the results of the clustering phase were used as initial conditions. These initial conditions included the number of states and their probability functions. Several values of a transition matrix were used for initialization of the algorithm. One of these values derived from the temporal segmentation of the previous phase, and the others was chosen at random. All the elements in the transition probabilities were non-zero, resulting in an ergodic model. The observations of the modeling were the firing rate vectors, thus the model was computed for continuous observations with one mixture each (Chapter 2 Section 2).

The computation of the full Hidden Markov Model was carried out in two ways:
4.5 Insertion of firing rate cross-correlations into the model

The multivariate Poisson process is taken as a maximum entropy distribution with i.i.d. Poisson prior, subject to pair-wise channel correlations as additional constraints, yielding Eq. 2.61.

The computation of the cross-correlation of the firing rate was carried out using an iterative algorithm based on Eq. 2.66 - 2.67. This algorithm was based on the first stage of the model, i.e. without transition probabilities. The algorithm consisted of the following steps:

1. Set $n = 1$

2. Choose initial values for the algorithm. These values are the $\lambda_i^{(n)}$ and $\sigma_{ij}^{(n)}$. 

---

Figure 4.2: An example of execution of a K-means algorithm for different number of states, using the same data.

1. Using the continuous observation Baum-Welch algorithm defined in Eq. 2.25 - 2.26 and 2.28 - 2.30 (Chapter 2 Section 6).

2. Using segmental K-means algorithm (described in Chapter 2 Section 6) using Eq. 2.12 - Eq 2.18.

The second method is less optimal than the first, but has been shown to produce similar results in other areas in which the HMM was used - automatic speech recognition, for instance [61].
Figure 4.3: An example of execution of a K-means algorithm for two sets of data vectors.

4. Compute the set $a_{ij}^{(n+1)}$ from Eq 2.67 using the values of $a_{ij}^{(n)}$ and $\lambda_i^{(n)}$.
5. Compute the set $\lambda_i^{(n+1)}$ from Eq 2.66 using the values of $a_{ij}^{(n)}$ and $\lambda_i^{(n)}$.
6. If the difference between the $a_{ij}^{(n)}$, $\lambda_i^{(n)}$ and $a_{ij}^{(n+1)}$, $\lambda_i^{(n+1)}$ is below a chosen threshold, stop here, and use the values $a_{ij}^{(n)}$, $\lambda_i^{(n)}$ as final values.
7. $n \rightarrow n + 1$;
8. Go to step No. 2.

It was found that the full computation of Eq 2.66 - 2.67 was not feasible because of the number of operations involved. The number of addition and multiplication operations at each iteration was of the order of $10^{10}$. Furthermore, no analytical shortcut was found, due to the Poisson probabilities (in contrast to the usual Gaussian distribution, in which the summation could be replaced by integration). Another problem was that most of the elements outside the diagonal of the clusters’ covariance matrix were very small. These two problems, when considered together,
CHAPTER 4. APPLICATION OF THE MODEL

suggested that only the strongest cross-correlation element could be computed. That element was chosen according to the values in the covariance matrix of each cluster.

The addition of the cross-correlation function, therefore, is limited to only one cross-correlation value per state.

4.6 Search for correlation between state sequences and events

One of the main hypotheses was that it is possible to predict the actual events (stimulus to the monkey or behavioral action of the monkey) from the states computed by the model. Therefore, after the data had been labeled by the HMM, a check for sequences in the temporal segmentation was carried out. This search, similarly to the process of constructing the whole model, was carried out in an unsupervised manner, i.e. the information regarding the events was not inserted into the search algorithm.

This search algorithm consisted of the following steps:

1. A specific event $E$ was chosen.

2. The sequence of 3 states which followed this event closely in one of the trials was chosen and marked as $SQ$.

3. All the occurrences of sequence $SQ$ were computed and marked on the data.

4. The previously found sequence locations were compared to location of event $E$ in the recording session, taking into consideration a possible delay of up to 100 ms between the event and the sequence. This comparison yielded the prediction of all other events of type $E$.

Several different events $E$ were used and for each of them several sequences of states $SQ$ were utilized. The sequence of states examined in this algorithm, was made of three states (as indicated in step No. 2). This is due to a preliminary examination, in which shorter sequences were used, but without any notable results.

4.7 Display tools

All the work described above was based on interactive tools, which enabled the research to check and display graphically the following information:

1. Spike trains of chosen cells.

2. Firing rate vectors of the data.

3. The clustering process (size of clusters, average divergence measure, etc.)

4. The centroids of the different models computed.

5. The temporal segmentation of the data according to different models.

6. The segments' length distribution.

In the research, a large amount of data was evaluated and these tools were used extensively in order to assess the results reached and to direct the research in the most promising direction.
CHAPTER 4. APPLICATION OF THE MODEL

4.8 Temporal cross-correlation in different states

The temporal cross-correlation function was computed for all of the pairs of cells in all their different states. This computation was performed as follows:

1. All the data from a recording session was segmented and labeled according to the full model (including transition probabilities).

2. All the spike trains found in the segments of a given state were collected together.

3. From the spike trains so selected, only segments with a length of more than 1000 ms were used to compute the cross-correlation functions.

4. The computation of the cross-correlation functions was done using an existing program with some slight modification. This program is based on renewal processes concepts [20]. According to these concepts, the inter-spike-intervals are positive uncorrelated random variables.

5. The results of this computation were gathered according to the model state, i.e. every figure included the correlograms of a specific pair of cells in all different states of the model. These figures were printed and scanned manually. A detailed description of the manual scan of the figures is presented in Chapter 6.

The cross-correlation function computed here is the correlation of the inter-spike time intervals (described in Chapter 3 Section 3), in contrast to the firing rate cross-correlation computed previously (described in Chapter 4 section 4). Another major difference between the two types of cross-correlation functions may be found in the time domain. The temporal cross-correlation can capture short-time cross-correlations (even in the range of few ms), whereas the time constants in the firing rate cross-correlations are much longer (in the order of hundreds of ms).
Part III

Results
Chapter 5

Temporal Segmentation

5.1 Multi-cell activity in different states

The basic results of the first stage of analysis (clustering of the data, as presented in Chapter 4 Section 3) were two sets of centroids for each recording session, one set for each behavioral mode of the session (“Go” mode vs. “No-Go” mode). These sets of centroids are the outcome of the K-means algorithm. A typical example of such centroids is presented in figure 5.1: 7 cells are shown (the cells are marked as 1 to 7 on the abscissa), and 6 centroids (marked as 1 to 6 on the ordinate). The different firing rates are represented as small bars, with the height of each proportional to the firing rate. Each firing rate vector computed from the data was associated with one of the centroids and thus these centroids induce a temporal segmentation of the data into states.

From this example it can be seen that the centroids differ from each other not merely due to a single cell activity, but due to the activity of several cells. For instance: the difference between centroid No. 1 and centroid No. 2 is in cells No. 1,5,6. In centroid No. 1 the firing rates of cells 1,5,6 are much lower than in centroid No. 2, whereas the firing rate of cell No. 2 stays the same in both.

This finding was consistent over all the recording sessions, i.e. in all the sets of centroids there were pairs of centroids which differed in changes of firing rates of more than one cell’s firing rate.

5.2 Consistency of clustering results

The clustering algorithm was repeated several times for each of the recording sessions (as presented in Chapter 4 Section 3). Each repetition differed in its initial conditions. Thus multiple sets of centroids were computed for each recording session.

It was found that all these multiple sets converged to a nearly identical centroid set with the same error measure. This fact was seen clearly when different recording sessions were taken, and also when clustering of data from the same recording day, but from different behavioral modes, was examined.

Two examples of centroids are presented in figure 5.2. This figure shows the result of the clustering phase with 6 centroids, with the firing rate of 7 cells each. All centroids are the results of measurements from the same recording session. Two sets of centroids (a and b) are the output
of data from the "Go" mode in this recording session. The other set of centroids (c) is the output of data from the "No-Go" mode.

The upper two centroid sets show a high similarity, whereas the lower set of centroids differs from them. An example of this difference can be seen in cell No. 5 in centroid No. 3. In sets a and b its firing rate is very low, whereas in set c its firing rate is very high. The differences between the sets from different behavioral modes exhibit some interesting phenomena:

1. Some of the centroids display almost identical firing rates (for example, centroids No. 5 and No. 2 are almost identical in all sets).

2. For the centroids that show differences between the behavioral modes, the difference is usually seen in the firing rate of one or two cells. In centroid No. 3, for instance, only the firing rate of cell No. 5 changes between different behavioral modes.

3. The average firing rate of cells does not change dramatically, but rather, the change is seen in the combination of firing rates in different centroids. Cell No. 6, for example, does not change its average firing rate between different behavioral modes, but the combination of high firing rate of cell No. 6 and low firing rate of cell No. 1 exists only in set c and not in sets a or b.
CHAPTER 5. TEMPORAL SEGMENTATION

Figure 5.2: An example of three sets of clustering results. The upper sets are from the same behavioral mode, the lower set is from a different behavioral mode.

The data presented above point to an interesting finding: the different models which characterize cells’ behavior in one behavioral mode were very similar, whereas models of different behavioral modes were very different from each other. This phenomenon holds for all the recording sessions examined. It enables us to reverse the process, i.e. to identify the behavioral mode of the monkey through analysis of the electrophysiological data from cells in the cortex.

5.3 Coherency of the clustering results

The next question to be examined was the accuracy of the restricted model (without transition probabilities and firing rate cross-correlations) in describing the data. The quantitative measure of accuracy, used in this model, was the average divergence measure of all firing rate vectors from their centroids. The degree of accuracy will henceforth be called the coherency of the model. The better
the model explains the data, the more coherent it is, and the smaller the measure of coherency.

A coherency measure is shown for one of the recording sessions, in figure 5.3. In this experiment two initial conditions were used, one consisting of 8 clusters, and the other of 15 clusters. For each of these conditions, 3 different overlapping segments of the firing rate vectors were used. These segments are sorted according to their length, i.e. the first segment is the shortest and the last segment is the longest. The first segment consisted of 20% of the data, the second segment consisted of 50% of the data, and the third segment consisted of 80% of the data. For each segment of the data, the clustering algorithm was initiated 6 times, each time from different starting conditions, i.e. different initialization of centroids. In figure 5.3, the number of the clusters is shown above the coherency measure. The three segments of the data used are marked on the abscissa by the numbers 1 to 3, and for each of these segments 6 measurements of coherency level are shown.

From this figure it can be seen that:

1. The different initialization of the centroids did not cause a large change in the results. Almost the same coherency level was reached from all 6 centroids. This result again demonstrates the consistency of the modeling process.

2. In all 3 segments of the data, and in the different number of clusters, there is a significant difference between the coherency in the "Go" mode compared with the "No-Go" mode.

3. The phenomenon described in the previous item is more prominent in the 8 clusters case than it is in the 15 clusters case.

In table 5.1 the behavior of this coherency is depicted for all of the 6 days that were checked. This table shows that the coherency measure differentiates between the different behavioral modes in all of the recording sessions. In 4 recording sessions the difference between the behavioral modes is significant (the significance level that was chosen here was a difference of more than 0.5 coherence). In three of them (bl16 and cl77 and bl39) significance is not reached, but the directionality is maintained.

<table>
<thead>
<tr>
<th>Recording Day</th>
<th>Minimum “Go” mode Coherency</th>
<th>Minimum “No-Go” mode Coherency</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>6.88</td>
<td>7.01</td>
</tr>
<tr>
<td>bl25</td>
<td>15.01</td>
<td>17.46</td>
</tr>
<tr>
<td>bl29</td>
<td>8.33</td>
<td>9.12</td>
</tr>
<tr>
<td>bl39</td>
<td>7.88</td>
<td>8.34</td>
</tr>
<tr>
<td>cl77</td>
<td>8.08</td>
<td>8.34</td>
</tr>
<tr>
<td>cl85</td>
<td>9.97</td>
<td>11.76</td>
</tr>
</tbody>
</table>

Table 5.1: The coherency of the best model for the “Go” mode and the “No-Go” mode at each of the recording sessions.

The findings presented above show that a difference exists between the coherence of spike trains...
of different behavioral modes. Based on this result, it is possible to predict the behavioral mode from the coherence measure for a given set of data. The prediction can be made in two ways:

1. Given models for different behavioral modes in the same recording session, and a set of spike trains from and unknown behavioral mode, it is possible to compute the coherence measure of this set and to relate it to one of the behavioral modes.

2. Given two sets of spike trains for different behavioral modes from the same recording session, and without appropriate models, it is possible to differentiate the set of spike trains from “Go” mode vs. “No-Go”. This differentiation is achieved by computing the optimal model for each set of spike trains, and the coherence of the set according to its model. The set of spike trains with greater coherence with respect to its model can be assumed to be the product of the “Go” mode.

5.4 States’ length distribution

The set of centroids computed by the model imposes a temporal segmentation of the data into states. An example of such a segmentation is presented in figure 5.5, in which the spike trains of 8 cells are shown. The temporal segmentation of the model is drawn over them (the vertical lines).
Consecutive basic time units (each consisting of 100 ms) belonging to the same state constitute a time segment. The indexes of these segments are shown above the spike trains, at the top of the figure. These indexes will henceforth be called the *labels* of the segments.

Figure 5.4 presents the accumulated distribution of segment lengths for both behavioral modes from one recording session. At each time point \( t \), the percentage of segments whose length is smaller than \( t \) is presented. For example, segments with a length of up to 1500 ms account for 79.4% of all the data in the “No-Go” mode, and for 70.8% of the data in the “Go” mode. It can be seen that the probability of finding longer segments in the “Go” mode is higher than in the “No-Go” mode, which means that the segments of the “Go” mode are longer than those of the “No-Go” mode. This phenomenon is consistent over all the segments of length 1000 ms and above.

This phenomenon was observed in 3 out of the 5 recording sessions, as seen in table 5.2. This table presents the percentage of segments with lengths up to 1500 ms, for both behavioral modes in all the recording sessions.
5.5 Effect of full HMM implementation

A full Hidden Markov Model is the term used here to describe the insertion of transition probabilities between states into the previously computed models. The computation of the full Hidden Markov Models of the data was done in two different ways (Baum-Welch algorithm and segmental K-means algorithm) as described in Chapter 4 Section 4. The results of these algorithms included the temporal segmentation of the spike train, and the likelihood of the data given the computed model.

5.5.1 Differences between the Baum-Welch algorithm and the segmental K-means

The results of the Baum-Welch algorithm and the segmental K-means were almost identical. They were identical in both result types, i.e. in the likelihood and the temporal segmentation. Table 5.3 presents typical likelihood values of one recording session as computed according to the following methods:

1. Clustering algorithm (K-means).

Figure 5.5: An example of a typical segmentation of spike trains of 8 cells.
2. The segmental K-means.

3. The Baum-Welch algorithm.

The likelihood values shown are the log likelihood per vector, i.e. \( N^{-1} \log(p_{\Lambda}(x)) \), where \( N \) is the number of firing rate vectors, \( \Lambda \) is the estimated model for the data, and \( x \) are the firing rate vectors.

<table>
<thead>
<tr>
<th>Session</th>
<th>K-means</th>
<th>segmental K-means</th>
<th>Baum-Welch</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>-0.88</td>
<td>-0.85</td>
<td>-0.85</td>
</tr>
<tr>
<td>bl25</td>
<td>-1.95</td>
<td>-1.90</td>
<td>-1.89</td>
</tr>
<tr>
<td>bl29</td>
<td>-1.03</td>
<td>-1.01</td>
<td>-0.99</td>
</tr>
<tr>
<td>bl39</td>
<td>-0.98</td>
<td>-0.97</td>
<td>-0.97</td>
</tr>
<tr>
<td>cl77</td>
<td>-1.01</td>
<td>-1.00</td>
<td>-0.99</td>
</tr>
<tr>
<td>cl85</td>
<td>-1.34</td>
<td>-1.33</td>
<td>-1.33</td>
</tr>
</tbody>
</table>

Table 5.3: The likelihood values achieved in the different training algorithms.

These likelihood values show a sustained behavior, i.e. the changes between the different computations is only minor compared with the likelihood values themselves. This behavior was seen in all the recording sessions. The computation of algorithms No. 2 and 3 was done on the same initial values reached in algorithm No. 1. Thus, it is possible to use the segmental K-means algorithm instead of the time consuming Baum-Welch algorithm, for this type of analysis.

### 5.5.2 Changes in temporal segmentation due to a full HMM

The main effect of the HMM on the temporal segmentation was the smoothing of the segment sequence, i.e. elimination of short segments and their absorption into longer segments. *Smoothing* is defined as the process of eliminating short, low-probability, state segments (usually 100-200 ms long) from the state sequence. This resulted from the insertion of transition probabilities into the model. The probability of short segments is therefore very small, and most of them are removed.
Thus, the main determinant of the segment label is the multivariate Poisson probability. The transition probabilities govern only the elimination of short segments, and not the labeling of segments.

An example of the differences between the clustering result and the full Hidden Markov Model can be seen in figure 5.6. Figure 5.6a shows the spike trains of 8 cells. On each of the spike trains, the firing rate is drawn. The temporal segmentation according to the Hidden Markov Model is depicted above the arrow (but should be extrapolated to the entire dot-display). The labels of states according to this segmentation appear above the graph. The segmentation according to clustering is depicted below the arrow (and should also be extrapolated to the entire dot-display). The labels of states according to this segmentation appear below the graph. Figure 5.6b is parallel to figure 5.6a, and presents the probabilities for being in each of the states depicted in figure 5.6a at each time point. The probability calculation was carried out based on the vector of firing rates at each time point. It can be seen that the temporal segmentation of the clustering is derived directly from the probability graph (see figure 5.6a below the arrow), whereas the segmentation of the full HMM results only in the smoothing of this graph (see same figure above the arrow).

The effect of a smoothing process can be seen in the two temporal segmentations of data in figure 5.6a. The state sequence in the clustering segmentation (see below the dot-display) is "1 5 1 4 1 5 2 4", whereas in the full HMM, the state sequence is just "1 5 4" (see above the dot-display). It can be seen that the first appearance of state No. 5, and the first appearance of state No. 4 are eliminated from the clustering segmentation, and both are replaced by state No. 1. This exchange of states is the result of the fact that states No. 5 and No. 4 are short segments (100 ms long each), with a low probability (as seen in the probability graph of figure 5.6b). State No. 1, on the other hand, has a higher probability, on average, in the whole region under discussion (state segment No. 1 according to the HMM segmentation).

It was found that this smoothing process is responsible for 96% of the differences between the HMM segmentation and the clustering segmentation. Smoothing, therefore, constitutes the strongest effect of the full HMM model. These findings were consistent over all the recording sessions.

### 5.6 Effect of insertion of firing rate cross-correlations into the model

The insertion of the cross-correlation coefficients into the HMM did not alter the temporal segmentation of the data. The vast majority of cross-correlation coefficients computed were zero, indicating that no pair-wise cross-correlation exists between the firing rate of different cells in a single cluster. Furthermore, in the rare instances when these coefficients were not-zero, their values were very small, and did not effect the temporal segmentation of the data in any way.

An example of a result of this calculation can be seen in figure 5.7. Three covariance matrices of 7 cells are shown, indicating the correlation between the vectors in each of three different states. The findings from these matrices are:

1. Almost all of the non-diagonal elements in each matrix are zero.
2. The largest absolute values that the elements in the covariance matrix reached are -0.20 and 0.12.
3. The covariance elements that hold non-zero values change their values in different states, i.e. no significant cross-correlation is indicated by these matrices.

In each matrix, the highest covariance value was chosen (marked in the figure), and the cross-correlation was computed between the firing rates of these cells (using the iterative algorithm described in Chapter 4 Section 5). In the example shown, the cross-correlation coefficients for all 3 matrices were identically zero. These values were reached after 3-4 iterations of the algorithm. These coefficients did not change the temporal segmentation at all.

5.7 Prediction of events based on state sequences

Our original hypothesis was that the sequences of states would be able to predict the events of the animal at the time of measurement. The results of the prediction test (described in Chapter 4 Section 6) are presented in table 5.4. This table includes the following information for each prediction trial:

1. The recording session code.
2. The number of sequence appearances in the data, for a specific sequence computed for this trial.
3. The number of times that the sequence successfully predicted the event under examination.
4. The total number of appearance for that event in the recorded data.

<table>
<thead>
<tr>
<th>Session</th>
<th>Sequence num</th>
<th>Correct predictions</th>
<th>Total Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>b116</td>
<td>243</td>
<td>21</td>
<td>281</td>
</tr>
<tr>
<td>b129</td>
<td>170</td>
<td>12</td>
<td>157</td>
</tr>
<tr>
<td>b129</td>
<td>123</td>
<td>10</td>
<td>78</td>
</tr>
<tr>
<td>cl77</td>
<td>327</td>
<td>32</td>
<td>403</td>
</tr>
<tr>
<td>c177</td>
<td>411</td>
<td>28</td>
<td>403</td>
</tr>
</tbody>
</table>

Table 5.4: Typical examples of prediction of events using states’ sequences.

This table shows that this hypothesis was not confirmed unequivocally from the data presented up to this point. That is, the simple sequences of states by themselves cannot give reasonable prediction of the events of the animal. However, some observations do seem to lend some support to the hypothesis. Figure 5.8 illustrates this point. Figure 5.8a and 5.8b represent dot-displays of the same 8 cells, at two different points of time during one recording session. In the two figures, the line marked Start refers to the same external cue (in this case, beginning of trial of a "Go" mode). It can be seen that the state sequence following this Start is identical at the two points of time. The sequence is "1 5 2".
This example and many other similar findings, suggest that a connection does exist between
sequences of states and external cues - but this connection had not yet been successfully captured.
Figure 5.6: An example of two temporal segmentations of the same data (a.) and all their clusters’ respective probabilities (b.).
### Figure 5.7: An example of covariance matrices of 7 cells in 3 different states.

<table>
<thead>
<tr>
<th>State 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
</tr>
<tr>
<td>-0.0503</td>
</tr>
<tr>
<td>-0.0160</td>
</tr>
<tr>
<td>0.0302</td>
</tr>
<tr>
<td>0.0080</td>
</tr>
<tr>
<td>-0.0171</td>
</tr>
<tr>
<td>0.0095</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
</tr>
<tr>
<td>0.1234</td>
</tr>
<tr>
<td>-0.1046</td>
</tr>
<tr>
<td>0.0277</td>
</tr>
<tr>
<td>0.0908</td>
</tr>
<tr>
<td>-0.0504</td>
</tr>
<tr>
<td>-0.0253</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
</tr>
<tr>
<td>-0.0821</td>
</tr>
<tr>
<td>-0.0095</td>
</tr>
<tr>
<td>0.0025</td>
</tr>
<tr>
<td>-0.0252</td>
</tr>
<tr>
<td>-0.0081</td>
</tr>
<tr>
<td>-0.0046</td>
</tr>
</tbody>
</table>

**Cells**
Figure 5.8: An example of two segmentations and two labelings of spike trains at different time points of the same recording session.
Chapter 6

Temporal Cross Correlations in different states

6.1 Working hypotheses

The previous chapter dealt with cross-correlations between firing rates of cells. In the present chapter, the temporal cross-correlation of spike trains of cells will be discussed. The firing rate cross-correlation was computed over a computation window of 500 ms width, whereas the temporal cross-correlation is computed in lags of single ms. The goal of the test presented in this chapter was to check the existence of differences between cross-correlation functions of this type in different states of the HMM, and thus, to try to connect the ideas underlying the cell-assembly hypothesis with the HMM based model.

A non-flat temporal cross-correlation function of two cells may be the result of one of the following possible situations:

1. Direct interaction between the two cells, as a result of a strong synapse.
2. Both cells are derived from a common input, which may be the hypothesized cell-assembly.

In the second case the cross-correlation function is expected to change in different states of organization of the network - different activations of cell-assemblies. These changes may occur in time constants of tens to hundreds of ms.

6.2 Cells with sufficient data

The computation of the temporal cross-correlation was done as described in Chapter 4 Section 8. This first stage of computation consisted of choosing the states whose length was above a certain threshold. In this test the threshold used was 1000 ms. As a result of this filtering, only part of the data could be used for further analysis. The percentage of data used is shown in table 6.1. The information included in the table is as follows:

1. The recording session code.
2. The total length of data in the recording session (in ms).
3. The length of all states used (in ms).
4. The percentage of data used from the total length.

<table>
<thead>
<tr>
<th>Session</th>
<th>Total Length</th>
<th>States Used</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>8004</td>
<td>3149</td>
<td>39</td>
</tr>
<tr>
<td>bl25</td>
<td>11333</td>
<td>2372</td>
<td>21</td>
</tr>
<tr>
<td>bl39</td>
<td>11854</td>
<td>8664</td>
<td>73</td>
</tr>
<tr>
<td>cl77</td>
<td>16512</td>
<td>4632</td>
<td>28</td>
</tr>
<tr>
<td>cl85</td>
<td>8562</td>
<td>2688</td>
<td>31</td>
</tr>
<tr>
<td>Total</td>
<td>56262</td>
<td>21505</td>
<td>38</td>
</tr>
</tbody>
</table>

Table 6.1: The percentage of data used for the temporal cross-correlation, from the total length of recorded data.

The first step in analyzing the pair-wise cross-correlation function is to check whether they have sufficient counts in their bins to warrant analysis. Figure 6.1 shows examples of two cross-correlograms. Correlogram b was considered as having sufficient data for analysis, whereas the amount of data in correlogram a was considered insufficient. Correlogram a, therefore, cannot be used for further analysis. The insufficiency of data is characterized by the following features:

1. Numerous bins do not include any counts at all.
2. The average number of counts in a bin is low.
3. The bins that include any counts do not form a continuous region.

Figure 6.1: Two cross correlograms that show the effects of insufficient data.
The lack of sufficient data can be explained by the following. The expected number of counts in a bin resulting from a cross-correlation of two stationary and uncorrelated spike trains is:

\[ E_{ij}[\text{Counts per Bin}] = \lambda_i \lambda_j T \Delta t \]  

(6.1)

where \( \lambda_i \) and \( \lambda_j \) are the firing rates of cells \( i \) and \( j \) respectively, \( T \) is the measurement time, and \( \Delta t \) is the bin width used in the correlogram. In figure 6.1 the measurement time is the same for both correlograms (352 Seconds) as is the bin size (5 ms). The difference can be seen in the firing rates of the cells: in correlogram b, the firing rates of the cells are 11.54, and 2.01 spikes-per-second, whereas in correlogram a the firing rates are 0.18, and 2.0 spikes-per-second. Therefore, the insufficiency of data is mainly the result of low firing rates of cells. Unfortunately, most of the well-isolated cells tend to have a very low firing rate (in the order of several spikes-per-second).

The problem of insufficient data in correlograms may cause the following effects:

1. Noisy correlograms may not reveal an inherent delicate temporal structure.
2. Departures from background in correlograms may be the outcome of noisy activity.

These effects may be crucial when testing the working hypothesis - for example, a functional connectivity that exists in one state may not be revealed in a different state, due to the insufficiency of data. Thus, insufficient data may insert artifacts into this test.

In order to overcome the problems described above, correlograms were used in further analysis, or rejected, on the basis of the following rules:

1. Correlograms in which most of the bins did not include any counts, and the rest of the bins included only a few counts (0-5), were not included. The rejection was based upon the average counts-per-bin, and the threshold was set at 0.2 counts-per-bin.

2. Correlograms would be used for further analysis if they included a continuous region (typically around delay 0) in which the number of counts was in the order of tens-per-bin.

3. If more than two correlograms between the same pair of cells in different states did not hold sufficient data, all correlograms between that pair of cells were excluded.

The second rule was added in order to include examples such as that presented in fig 6.2. The correlograms shown in this figure, are computed for the same condition, and differ only in the bin size. In a, the bin size is 5 ms, whereas in b the bin size is 25 ms. According to first rule described above, this correlogram should not be used for further analysis, but a significant departure from background noise can be seen, especially in the larger bin resolution (example b). This correlogram shows a significant functional connectivity between the cells, and rule No. 2 enables us to make use of it.

The information about the correlograms that include sufficient data to warrant further analysis, is shown in table 6.2. The information included in the table is as follows:

1. The recording session code.
2. The number of well-isolated cells used in each recording session.
3. The number of cross-correlograms computed for each of the different states of the Hidden Markov Model. This number is $N(N - 1)/2$, where $N$ is the number of cells used in the recording session under examination.

4. The number of cell-pairs which included sufficient data. This number may be taken as an indication of the activity of cells in the recording session under examination.

<table>
<thead>
<tr>
<th>Session</th>
<th>Cell Num</th>
<th>Total Cross Num</th>
<th>Sufficient Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>8</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>bl25</td>
<td>8</td>
<td>28</td>
<td>22</td>
</tr>
<tr>
<td>bl39</td>
<td>6</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>cl77</td>
<td>6</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>cl85</td>
<td>8</td>
<td>28</td>
<td>25</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>36</strong></td>
<td><strong>114</strong></td>
<td><strong>99</strong></td>
</tr>
</tbody>
</table>

Table 6.2: The number of sufficient data cross-correlograms, among the cross-correlograms of the chosen cells.

This table shows that most of the correlograms hold sufficient information to achieve results: although the firing rate is usually low, there is sufficient data to justify further analysis. Furthermore, it was found that most of the data used showed a high number of counts-per-bin, and the number of correlograms included due to rule No. 2 (regarding the continuous region of counts) was small.
6.3 Flat correlograms vs. functional connectivity

After defining which correlograms hold sufficient data, the next step was to examine the data in these correlograms. Typical examples of correlograms are shown in figure 6.3: example a. shows a flat correlogram, which implies that the activity of the cells is uncorrelated. In example b, however, there is a rise in activity around zero delay - which implies that the cells tend to fire simultaneously. The strategy consists of a search for departures from the background of the cross-correlation function.

![Correlogram Examples](image)

Figure 6.3: An example of three correlograms. a: flat correlation function, b: nonzero correlation, c: questionable nonzero correlation

The difference between a flat function and a departure from background may not always be as clear as that between examples a and b. Example c. demonstrates such a case: this correlogram shows a small departure from background at zero delay. This departure is not as clear as in example b, but cannot be ignored. In order to differentiate between these two types, and mark example c.
as a flat correlogram, significance lines were used.

The significance lines are drawn on the correlograms. They indicate the degree of confidence in the claim that the number of counts in a bin is not the result of a counting process whose mean is the average counts in bins. Based on this rule, example b in figure 6.3 is considered a clear deviation from flat correlation, whereas in example c the deviation is very small. Example c cannot be considered a flat correlogram when compared with correlogram with clear departure from background, but neither can it be counted as a clear departure from background when compared with flat correlograms.

This classification is not a strict one, and takes into consideration the shape of the correlogram being checked. This can be seen in figure 6.4, where the correlogram depicted would not be considered a non-flat correlogram, despite the fact that in the bins (marked by arrows) the counts pass the significance line. These crossings are not significant, and are not continuous. These crossings are due to the fact that the deviance from the line is very small and local. In addition, these confidence lines are based on 1% confidence level, and there is a possibility, albeit a small one, that a purely random process could cross that confidence line.

Figure 6.4: An example of a correlogram which is basically flat although the cross-correlation function crosses the significance lines.
CHAPTER 6. TEMPORAL CROSS CORRELATIONS IN DIFFERENT STATES

The process of deciding upon the non-flat correlograms can be defined as follows:

1. Choose all the correlograms that hold sufficient data.

2. From the previous subset, choose all the correlograms that show a departure from background and cross the significance lines.

3. From the previous subset, choose all the correlograms for which the departure from significance lines is either continuous and lasts for at least several tenths of ms, or the departure is very strong, i.e. much higher than the significance line.

The number of non-flat correlograms computed by that method in the recording sessions is shown in table 6.3. The table holds the following information:

1. The recording session code.

2. $M$ - The number of states used in the modeling of each recording session

3. $N$ - The number of sufficient data pairs of cells, as found in the previous analysis. The total number of correlograms checked is $M \times N$, and is shown in parenthesis in this column.

4. The number of pairs of cells in which non-flat correlograms were found, and the number of actual correlograms found (out of $N \times M$ total correlograms). The latter number is shown in parenthesis. These numbers were computed separately for the "Go" mode and for the "No-Go" mode.

5. The percentage of pairs of cells that show non-flat correlogram and the percentage of non-flat correlograms from the total number of correlograms (shown in parenthesis). These percentages were computed separately for the "Go" mode and for the "No-Go" mode.

<table>
<thead>
<tr>
<th>Session</th>
<th>States ($M$)</th>
<th>Sufficient Data ($N$)</th>
<th>Non-Flat</th>
<th>Non-Flat Percentage</th>
<th>Go</th>
<th>No-Go</th>
<th>Go</th>
<th>No-Go</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>6</td>
<td>28 (168)</td>
<td>5 (15)</td>
<td>1 (3)</td>
<td>17.9 (8.9)</td>
<td>3.6 (1.8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bl25</td>
<td>7</td>
<td>22 (154)</td>
<td>4 (7)</td>
<td>2 (3)</td>
<td>18.2 (4.5)</td>
<td>9.1 (1.9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bl39</td>
<td>7</td>
<td>13 (91)</td>
<td>4 (12)</td>
<td>1 (1)</td>
<td>30.8 (13.2)</td>
<td>8.5 (1.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cl77</td>
<td>6</td>
<td>11 (66)</td>
<td>7 (21)</td>
<td>2 (5)</td>
<td>63.6 (31.8)</td>
<td>18.2 (7.6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cl85</td>
<td>6</td>
<td>25 (150)</td>
<td>6 (16)</td>
<td>5 (15)</td>
<td>24 (10.7)</td>
<td>20 (10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
<td>99 (629)</td>
<td>26 (71)</td>
<td>11 (27)</td>
<td>26.2 (11.3)</td>
<td>11.1 (4.3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: The number of non-flat correlograms among the sufficient data correlogram

It can be seen from this table that a non-negligible number of correlograms reveal a tendency to deviate from a flat cross-correlation. This fact is crucial for the next steps, because if all the correlograms had shown flat correlations there would be no point in checking the changes between states. The results presented in table 6.3 show that a functional connectivity exists between pairs
of cells in specific states of the model. This result may be due to the modeling of the data, or it may be the result of a prominent functional connectivity that exists throughout the whole recording session.

This table also shows that the percentage of non-flat correlograms is much higher in the “Go” mode than in the “No-Go” mode. In the “Go” mode 11.3% of the correlograms were non-flat, whereas in the “No-Go” mode only 4.5% were non-flat. This phenomenon is also seen in the percentage of pairs of cells that show any correlation. In the “Go” mode they constitute 26.2%, whereas in the “No-Go” mode they constitute only 11.1%.

### 6.4 Types of functional connectivity

The types of departures from background of the cross-correlation function that were found in this analysis, can be seen in figure 6.5. There were 4 types:

A. *Narrow* peak confined to one side of the origin, in this example to the positive side.

B. *Narrow* peak in the correlogram - straddling the zero delay.

C. *Narrow* trough confined to one side of the origin, in this example to the negative side.

D. *Wide* peak straddling the zero delay in the correlogram.

The terms *wide* and *narrow* refer to the departure time. By *wide* we mean a peak or a trough that lasts more than 200 ms and by *narrow* - that the time constants involved are 10-100 ms. Types B and D are usually referred to as *shared* or *common input*.

After computing all the non-flat correlograms, they were classified according to these four types. Table 6.4 shows this classification:

<table>
<thead>
<tr>
<th>Session</th>
<th>Type A</th>
<th>Type B</th>
<th>Type C</th>
<th>Type D</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>b116</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>b125</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>b139</td>
<td>1</td>
<td>1</td>
<td></td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>c177</td>
<td>2</td>
<td>16</td>
<td>3</td>
<td>5</td>
<td>26</td>
</tr>
<tr>
<td>c185</td>
<td>3</td>
<td>13</td>
<td>15</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>11</td>
<td>39</td>
<td>3</td>
<td>45</td>
<td>98</td>
</tr>
</tbody>
</table>

Table 6.4: The classification of non-flat correlograms into four types of functional connectivity

This table shows that 46% of the non-flat correlograms showed wide peaks, and 40% showed narrow peaks. Thus these two types constitute 86% of the total number of non-flat correlograms. It should be noted that no troughs straddling zero delays were found in this experiment.
Figure 6.5: An example of different types of functional connectivity

6.5 Modification of pair-wise cross-correlation functions between states

The Modification of Pair-wise Cross-correlation function between States (MPCS) is defined as all pairs of cells that have different types of functional connectivity in different states of the model, i.e., in one of the states one type of functional connectivity is observed, whereas in other states a different functional connectivity is seen. When computing all the MPCS, it was possible to use the flat type, i.e., an MPCS event would be counted if in one of the states a type C cross-correlation was observed, and in the other a flat correlation was seen. In any case of doubt regarding the existence of a functional connectivity, the criterion was as strict as possible. The MPCS would not be counted if either:

- It was not clear that a functional connectivity existed.
• It was not absolutely clear that the correlogram was flat.

A typical example of such a situation is shown in figure 6.6. In this example, the number of states of the HMM is 7, so correlograms numbered from 0 to 6 of the same pair of cells (No. 5 and No. 8), are depicted. Each of these correlograms was computed for the spike trains of the matching state. In the correlogram of state No. 1, a clear, non-flat, cross-correlation function exists between the pair of cells, whereas in the correlogram of state No. 6 the correlogram is of the flat type. The cross-correlation function in state No. 1 is of a common input type, i.e. the rise of the cross-correlation function above the significance line occurs on both sides of zero time. The size of this rise is in the order of several hundreds of ms. In the correlograms of states 0, 2 and 3, a similar, common input phenomenon is shown. In state 3, however, the departure from background and crossing of the significance line are much weaker compared with states 0-2. On the other hand, in the correlograms of states 4, 5 and 6 no such departure is seen.

![Correlograms of different states](image)

Figure 6.6: All the correlograms of a pair of cells in different states.

Some other phenomena should be noted in this example:

1. The firing rate of the cells is very low (less than a spike-per-second for most of the states), but still, a very persuasive correlation function can be seen. For example the firing rates of cell No. 8 in states No. 0-4 are 1.15, 0.63, 0.7, 0.64, 0.83 respectively. The firing rates of cell No. 5 in states No. 1, 2, 4 are 0.83, 1.05, 0.57 respectively.
2. Correlation shapes may change between two states even when almost no changes exist in the firing rate of the pair of cells. For example, in state No. 1, cell No. 5 fires at a rate of 0.83 spikes per second, whereas in state No. 4 the average firing rate is 0.57. For cell No. 8, the firing rates are 0.63 spikes per second and 0.83 spikes per second respectively. The relationship between MPCS and the firing rate in different states will be discussed later.

3. A rise is seen on the positive end of the correlogram of state No. 5. A possible explanation of this rise is the tendency of cells to intensify their firing rate towards the end of the state. Such an explanation, however, goes beyond the scope of the current research.

In table 6.5 the results of this stage of analysis are shown. The table presents the following information:

1. The recording session.
2. The total number of pairs of cells with sufficient data.
3. The number of MPCS occurrences in the “Go” mode.
4. The number of MPCS occurrences in the “No-Go” mode.

It can be seen that a non-negligible number of cell pairs show occurrences of MPCS, and that this percentage is much higher in the “Go” mode compared to the “No-Go” mode.

<table>
<thead>
<tr>
<th>Session name</th>
<th>Sufficient Data</th>
<th>“Go” mode</th>
<th>“No-Go” mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>28</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>bl25</td>
<td>22</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>bl39</td>
<td>13</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>cl77</td>
<td>11</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>cl85</td>
<td>25</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>99</strong></td>
<td><strong>25</strong></td>
<td><strong>10</strong></td>
</tr>
</tbody>
</table>

Table 6.5: The number of MPCS occurrences - classified into “Go” trials and “No-Go” trials, and compared with the total number of sufficient data cell-pairs.

### 6.6 Contamination of spike train as a cause of MPCS

A possible artifact regarding the occurrences of MPCS may be due to contamination of spike trains from different recorded cells. This contamination may result in a cross-correlation functions which is a mixture of several correlation functions, whose relative weight change over time. These errors may result an inaccurate MPCS. This contamination may be the result of two factors:

1. Insufficient isolation between cells: the recorded spike trains of the cells consisted of the action potentials of two or more neurons. This insufficient isolation was discussed in detail in Chapter 3 Section 4.
2. Cells recorded from the same electrode: the two cells were recorded from the same electrode, thus the probability of the cells contaminating each other is higher than usual.

Each of these factors was checked, by going over all the MPCS occurrences found and checking the supposed influence of the factors compared with the MPCS occurrences found without their presence.

6.6.1 Insufficient isolation between cells

Table 6.6 shows the isolation quality of cells participating in all of the MPCS occurrences (both "Go" mode and "No-Go" mode). This isolation quality is the score the cells were given during the recording session (Chapter 3 Section 4), and is divided into 3 classes:

1. Both cells were well-isolated neurons, i.e. both of them are of type A or B.

2. One of the cells is a well-isolated neuron, whereas the other may be a mixture of two neurons. In score terms, this means that one of the cells is of type B and above, and the other is of type C and above.

3. Both of the cells are mixtures of two neurons, i.e. both of them are of type C and below.

<table>
<thead>
<tr>
<th>Session</th>
<th>B - B pairs</th>
<th>B - C pairs</th>
<th>C - C pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>bl25</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>bl39</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cl77</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>cl85</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16</strong></td>
<td><strong>16</strong></td>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

Table 6.6: Classification of MPCS occurrences for each recording session into three groups: A or B types, B or C types, and all the other types.

Table 6.6 shows that 45% of the MPCS occurrences (16 out of 35) are from well-isolated neurons, 91% of them (32 out of 35) are from at least one isolated neuron, and only 9% (3 out of 35) are from cells that are both mixtures of two neurons. Thus the vast majority of MPCS occurrences were from pairs of cells which are well-isolated cells, and the probability of contamination is very small.

6.6.2 Cells recorded from the same electrode

As explained previously (Chapter 3 Section 4), cells recorded from the same electrode have a higher risk of cross-contamination, thus creating artifacts in their cross-correlation function. Therefore,
the following test was conducted: the number of MPCS occurrences from the same electrode was compared with the number of MPCS occurrences from different electrodes. This check is presented in table 6.7 and includes the following information:

1. The recording session code.

2. The number of MPCS occurrences that were computed for cells from:
   (a) The same electrode in the “Go” mode.
   (b) Different electrodes in the “Go” mode.
   (c) The same electrode in the “No-Go” mode.
   (d) Different electrodes in the “No-Go” mode.

3. The total number of pairs that were recorded from the same electrode. This number is the same for “Go” and “No-Go” modes.

4. The total number of pairs that were recorded from different electrodes.

<table>
<thead>
<tr>
<th>Session</th>
<th>Go:same</th>
<th>Go:diff</th>
<th>No-Go:same</th>
<th>No-Go:diff</th>
<th>All:same</th>
<th>All:diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>bl16</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>bl25</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>bl39</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>cl77</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>cl85</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>20</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>104</td>
</tr>
</tbody>
</table>

Table 6.7: The number of MPCS occurrences from the same electrode and from different electrodes, in the 2 behavioral modes.

This table shows clearly that most of the MPCS were the result of cells recorded from different electrodes - 80% in the “Go” mode (20 out of 25) and 100% in the “No-Go” mode (10 out of 10). The total percentage of MPCS recorded from different electrodes is 86% (30 out of 35). Furthermore, the total number of pairs of cells recorded from different electrodes used in this experiment was 91% (104 out of 114). Thus, the percentage of MPCS occurrences recorded from different electrodes is very similar to that of the percentage of total number of cell pairs recorded from different electrodes. This result enables us to disregard the possibility that the MPCS occurrences were an artifact of cells recorded from the same electrode.

### 6.7 Types of MPCS

Figure 6.3 shows one type of MPCS: the transition from a flat correlogram a to a common input correlogram b. There are many types of such MPCS, each with its biological explanation. The
MPCS phenomena may be of a complicated type, as shown in figure 6.7. In this figure, several types of correlograms between the same pair of cells are shown in different states. The states are numbered from 0 to 5. The numbers of the first 3 states (0 to 2) appears below the respective correlograms. For states 3 to 5, the number appears above the respective correlogram. The different types are as follows:

1. Flat correlograms, in states No. 2, 4 and 5
2. Wide peaks - common input correlograms in state No. 3
3. Narrow peaks - common input correlograms in state No. 3.
4. Narrow peaks - one sided correlograms in states No. 0 and 1.

Figure 6.7: An example of 6 correlograms showing different functional connectivities.

Even within a single correlogram, several phenomena may be observed - such as a combination of a wide common input with a narrow one riding over it, as can be seen, for instance, in states No. 1 and 3 in fig 6.7.
The following section lists the types of MPCS phenomena found in the data:

1. Wide to flat: An example of that type can be seen in figure 6.3, where a. shows a flat correlation, and b. shows a wide common input. This type was one of the two most common types.

2. Wide to narrow: An example of that type can be seen in figure 6.8, where a. manifests a wide common input, and b. a narrow common input.

3. Narrow trough to flat: An example of that type can be seen in figure 6.9, where a trough is seen in b., and a flat correlation in a. This type was very rare - there was only one occurrence of a convincing inhibitory type of cross-correlation function, and therefore only one occurrence of inhibitory to flat type.

4. Narrow to flat: An example of that type can be seen in figure 6.10, where b. shows a narrow correlation, and a. shows a flat one. This type was also one of the two most common types.

5. One sided to double sided: An example of that type can be seen in figure 6.11, where a. shows a one sided peak, and b. shows a double sided peak. This type was also rare.
6.8 Changes in firing rates of MPCS

The basic parameters for the model presented here are the firing rates of all the cells. Different states of the model are deduced from differential firing rates. Therefore, we can expect that most of the pairs of cells will have different firing rates in different states. The possibility of pairs of cells with almost the same firing rate in different states does, however, exist. The reason for this is that the modeling is carried out using all the cells and the difference between states does not require that all of the cells should have different firing rates.

From the above, we would expect most of the MPCS occurrences to be between cells with different firing rates. We are faced with the following question: for a given pair of cells, are their different firing rates the only dominant factor, or can MPCS phenomena occurrences also be found between cells with the same firing rate?

In the example presented in Figure 6.3, although the firing rates of both cells do not change dramatically between states, the correlograms differ from each other. Another example of this type of situation can be seen in the example in figure 6.12, in which the correlograms of one pair of cells are shown in two different states. It can be seen that both correlograms do not show any departure from background - just a flat behavior, whereas the firing rate of the cells tends to change dramatically between the different states. The firing rate of cell No. 8 is 2.34 spike-per-second in 6.12a., compared with 5.96 spikes-per-second in 6.12b., and cell No. 13 changes its firing rate from 9.56 spikes-per-second in a. to 39.44 spikes-per-second in b..

Both examples show that the dependence of the MPCS phenomena upon changes in the firing rates is not an absolute one. Results of this check are summed up in table 6.8. This table shows the partitioning of all the MPCS events into two groups:

1. All the MPCS in which the firing rate of the cells did not change dramatically. The threshold used in this computation was that the firing rate would stay within boundaries of ± 50%.

2. All the MPCS in which the firing rate did change dramatically.

These figures were compared with the percentage of changes in firing rate in the total number of cross-correlograms (referred to as non-MPCS).
It can be seen from the table that the relative size of the 2 groups (no change in firing rate vs. dramatic change in firing rate) remains the same both for the MPCS data and for the non-MPCS data. From these results we can conclude that changes in firing rate alone cannot explain the phenomenon of the MPCS occurrences: the same percentage of changes in the firing rate exists between cells for which no MPCS was found.
Figure 6.11: An example of one sided to double sided functional connectivities.

Figure 6.12: An example of a dramatic change in firing rate without any change in the functional connectivity.
Part IV

Discussion
Chapter 7

Discussion

7.1 Hypotheses of the research

The aim of the work presented here was to construct a model relating the hypothesized cell-assemblies and the attractors introduced in ANN, with the spike trains of several cells recorded simultaneously. The model used was a Hidden Markov Model, whose parameters were the firing rate vectors of several cells recorded simultaneously. Computing such a model yields a temporal segmentation and labeling of the spike trains of the cells.

There were two working hypotheses. It was hypothesized that a connection exists between the states of the model and the behavioral events of the animal. This hypothesis received partial confirmation: the model was able to predict the behavioral task performed by the animal at the time of recording ("Go" vs. "No-Go" paradigm), but was unable to reveal a strong connection between the states of the model and the events of the experiment.

It was further hypothesized that a connection exists between the states defined above and the functional connectivity between cells, i.e. the functional connectivity between pairs of cells changes in different cognitive states. This hypothesis was confirmed. It was found that the functional connectivity, as reflected in the pair-wise cross-correlation, tended to change in different states of the model. The computation of cross-correlations yielded an impressive number of such changes. Furthermore, in almost all the occurrences of functional connectivity between cells, such a change was found.

7.2 Interpretation of results

It is important to note that, in contrast to previous research quoted in the literature, all the results shown here were reached by modeling the data without averaging over spike trains. The data was treated as a point-wise process, and no assumptions were made about the location of events in the model.

7.2.1 Consistency of the model

The results of the computations of the Hidden Markov Model consisted of the parameters of the different states, their transition probabilities etc. These results were consistent: no matter what the
initial conditions of the model, the parameters of the model always converged to a nearly identical result. This consistency was observed in all the stages of computation: the clustering of the data, the formation of a full HMM, and the different training algorithms used. It seems to us impressive that the stochastic output of a complicated network, when sampled randomly by only 6-8 of its units, gives such consistent results.

This consistency implies that the model succeeds in capturing a basic structure underlying the data. Finding such an inherent structure is the first step towards building a model which correctly describes the process of generation of the data, and which enables us to make predictions based on the data. The assumed underlying structure is a set of cognitive states, with the cells altering their firing rates according to the different cognitive states.

7.2.2 Multi-dimensionality of the model

The firing rates in the different states of the model were found to differ from each other due to the activation not of one cell, but of several cells. This result implies that the formation of states is truly a vector process. Furthermore, this result shows that the multi-dimensionality is another inherent feature of the data. A relation would seem to exist between the firing rates of the cells, and this relation should be sought in a higher dimension of correlation, i.e. in a dimension higher than just 2 or 3. The consistency and multi-dimensionality of the process confirm our basic assumptions, and encourage the use of models based on simultaneous activity of cells. They also emphasize the importance of simultaneous recordings of several cells.

7.2.3 Differences between behavioral modes

The computed model was used to differentiate between the behavioral modes of the monkey ("Go" vs. "No-Go" paradigms). There are three non-overlapping aspects to this differentiation:

1. The differences between the firing rates of the centroids in the two behavioral modes.
2. The difference in coherency of data from the two behavioral modes.
3. The differences in states' length distribution between the two behavioral modes.

The differences between the firing rates in the different behavioral modes had two important characteristics: first - the two differed only in some of the centroids' firing rates, and second - that difference was multi-dimensional, i.e. the differences between those centroids appeared in more than one cell's firing rate. The first difference may be explained by the assumption that in the two behavioral modes, some cognitive states of the cortex are similar, and some differ. The second difference justifies once again the use of multi-dimensional models for the analysis of the data.

The coherency of the data varied across the models of each behavioral mode, and in the majority of the recording sessions, this difference was a significant one. This would seem to indicate that the coherency measure also captures some basic principle inherent in the data. The difference in coherency may have a biological explanation: in the "Go" mode, the animal may need to pay more attention during the delay time, whereas the other mode ("No-Go" mode), the animal need not pay as much attention to the task. In other words, the "No-Go" mode requires less concentration than the "Go" mode. In the former mode, therefore, the accumulative response of the cells in the
area responsible for this task is less coherent than in the latter. As a result, the model is better able to describe the activity in the cortex for the "Go" mode than for the "No-Go" mode.

The difference found in the states’ length distribution suggests that the different behavioral modes are modeled quite differently. As in the case of the coherency measure, this fact may be explained in a similar way by differences in required concentration.

7.2.4 Different models used in the research

There were several stages and modeling techniques aimed at achieving temporal segmentation of the data:

1. The clustering of firing rate vectors using the K-means algorithm.

2. The computation of HMM of firing rate vectors carried out by segmental K-means.

3. The same computation as in stage 2 but with the Baum-Welch algorithm.

4. The computation of HMM of firing rate vectors and pair-wise cross-correlation of the firing rates.

These techniques were presented in the order of their use in the analysis. The output of each of them served as the initial condition for the next stage. An exception was stage 3, which used the output of stage 1.

The results achieved by these different models were very similar both for the temporal segmentation of the data and for the likelihood of the data according to each of these models. This similarity exhibits the following phenomena:

1. The segmentation of the data is mostly governed by the probability of a firing rate vector being produced by a specific state. Thus, the influence of the transition probability between states (i.e. smoothing the state sequence) was of only secondary importance.

2. The modeling of the data may be classified together with other classical HMM problems, in which the difference between the Baum-Welch algorithm, the segmental K-means and the clustering of data is only a minor one. The differences between those methods have been discussed theoretically, in depth, elsewhere [48]. Those authors claim that under certain conditions "the increase in likelihood achieved by the Baum and the segmental K-means algorithms, is minor".

3. The insertion of pair-wise cross-correlations into the model did not alter the temporal segmentation at all, due to the fact that the cross-correlation coefficients computed were either zero or as close to zero as to make no difference in the modeling.

The lack of influence of the firing rate cross-correlation coefficients on the temporal segmentation may be explained by the fact that these coefficients are only the second order correlations between the cells. The first order cross-correlation was captured by the multi-cell firing activity in different states. The vectorial process of clustering the data captures the correlation between the cells, i.e. two highly correlated cells fire in a correlated way in different states of the model.
7.2.5 Prediction of events by the model

While the model successfully predicted the behavioral mode of the monkey, it was less successful in predicting the events from the sequences of states. Some connection was found between the two: the data showed that near the same event, the same sequence of states repeated itself several times. However, this connection was not strong enough to form the basis of a prediction procedure.

A possible explanation of this result is that we attempted to predict the activity of a network consisting of tens of thousands of cells, from only six to eight cells randomly chosen from that network. While this proved sufficient for predicting the behavioral mode, it may have been insufficient for the present task. A supervised algorithm may have enabled us to successfully predict the events, but the ability to model all of the data (as was done in this work) would then have been lost. In the works on supervised HMM [64, 63], it was found that such prediction is attainable under the following conditions:

1. The spike trains used are taken only from the vicinity of the stimulus predicted.
2. The training algorithm is done in a supervised manner.
3. The cells used in the modeling show a prominent response to the external stimulus under investigation.

The reported success in prediction of events, together with the limited connection revealed in this work, raise the possibility of predicting the events of the experiment in an unsupervised manner. In order to produce such results, one should keep in mind that the sequence of states computed from the model may be the outcome of a stochastic process defining the events. Thus, a more sophisticated model may be required to capture the underlying structure of such a process. This model might be able to take into consideration the probability functions of each state, its length, etc.

7.2.6 Changes in functional connectivity between states

The temporal segmentation of the data inspired by the model revealed different functional connectivities in different states. These findings confirm the second hypothesis and imply that the recorded cells may participate in different assemblies and may be activated under different conditions at different times, i.e. in different cognitive states of the small neural network recorded. Changes in functional connectivities between different behavioral events have been found in previous research [10, 9]. The novelty in the present work is that these changes in functional connectivities were discovered by the model directly from the data, without making use of the events of the experiment.

The check carried out showed clearly that in almost all cases in which functional connectivity existed, it changed in different states of the model, i.e. almost any pair of cells that showed a non-flat correlation function, also showed a change in functional connectivity between states of the model. It should be noted that the number of pairs of cells showing correlative behavior was far from negligible: 25% of the pairs of cells showed a state dependent non-flat correlation function.

This last result gains additional importance from the fact that one of the common types of changes in functional connectivities (MPCS), was a change from a narrow correlation to a flat one (type No. 4 in Chapter 6 Section 7). The importance of this finding derives from the fact that the
modeling of the data was based on firing rate vectors computed in windows of hundreds of \(ms\), and the narrow correlation may be in the order of only a few \(ms\).

Certain properties of this work strengthen the results:

1. The selection of pairs of cells was as strict as possible (see Chapter 6 Section 2).

2. The decision on changes in the functional connectivity (MPCS) was as strict as possible (see Chapter 6 Sections 3 and 5).

3. The possibility that the MPCS was an artifact of the recording method (detection of several cells recorded from the same electrode) was eliminated as described in Chapter 6 Section 6.

4. There was no direct relationship between the MPCS and the firing rates of cells (see Chapter 6 Section 8).

The last point was observed in two ways:

1. The connection between dramatic changes in the firing rate and the existence of MPCS was quantitively examined, and no such connection was found. The MPCS cannot, therefore, be explained solely by changes in firing rates.

2. Examples of MPCS occurrences, where the firing rates of cells are similar, were found.

The examples found in the second type of observation may be caused by changes in the activity of other cells in the network, which serve as input to the measured cells. This explanation suggests that the functional connectivity also expresses the correlation between the inspected cells' activity and the activity of other cells in the neural network.

The fact that a model trained with long duration firing rate, was able to reveal changes in the short-time pair-wise cross-correlation, implies that both of these parameters are a reflection of an underlying computation process existing in the cortex.

### 7.3 Limitations of the model

The basic parameters of the model were the firing rates of the cells. As a consequence, the exact temporal patterns of the spike trains of several cells simultaneously recorded, were not taken into account. The use of firing rates as parameters necessitated choosing one computation method. In this work, the method chosen was the averaging of the number of spikes in a sliding window of 500 \(ms\) width, and with a a transition of 100 \(ms\). This method has several drawbacks:

1. The overlapping estimation windows result in the insertion of dependencies between adjacent firing rate samples.

2. The width of the window (500 \(ms\)) may be too long for the measurement of instances of short-time dynamics.

3. The window is of a constant width, and does not adapt itself to the data being modeled.
However, the method was chosen for it's simplicity, as this work was a novel one, the aim being to form a more advanced model based on the knowledge gathered in this experiment.

The Markovian assumption underlying the model, is that the transitions between states depend only on the last state visited. This assumption may not always hold, even according to the cell-assembly hypothesis. The insertion of this assumption into the model is done in order to simplify the computation of the different questions regarding the parameters of the model. This assumption has been extensively used in the past in other models for which it was known not to hold completely, with good results.

No effort was made to choose cells especially suited to the model. The cells used were chosen only according to criteria which were independent of the model, such as their isolation scores in the recording session and their stationarity during the recording session. Other parameters, such as the responsiveness or the firing rates of cells, were not used. Information concerning the temporal cross-correlation between the cells chosen in the research existed. However, no use was made of it before the computation of the temporal cross-correlations of different states was completed. It is interesting to note that although no use was made of this information, the results reported here were reached in all the recording sessions and under all conditions. The strict adherence to independence of the process of choosing cells for analysis, and the process of modeling, strengthens the validity of the findings of the experiment.

7.4 Conclusions and suggested further work

The main question addressed by this work concerns the way that information processing is carried out in the cortex. This work was based on the assumption that this processing may best be described by a Markovian state machine, whose output is the spike trains of cells. It seems to us that the main contribution of this work is that this modeling succeeded in differentiating meaningful states in the brain, thus giving support to the assumption stated above. Furthermore, analysis of the results of this work suggest that a connection exists between the computed states of the model and the cognitive states of the brain. These states are characterized by internal consistency, and carry significance beyond their place in the model. This significance manifested itself in two ways:

- The model was able, within certain limitations, to predict the behavior of the animal from its computed states.
- The model originally dealt with the estimated parameters of processes which were presumed to take place in the cortex. The findings, however, seem to reveal underlying, primary, processing units in the cortex.

This suggests the possibility that the neural computation is inherently a complicated process, which can only be glimpsed through different measurable parameters. These parameters may be the averaged firing rate of cells, the short-time cross-correlation, the firing patterns emerging from synchronous activity and so on. A better understanding of the neural computation can only be achieved by combining all these parameters in a united framework, and modeling them together. The Markovian state machine offers such a possibility.

We see the modeling used in this work as the first step towards more sophisticated modeling of spike trains. Such modeling, used in future research, would open up new possibilities:
1. Building the models in a hierarchal ways, i.e. hierarchal clustering of data and hierarchal hidden Markov modeling. An example of such modeling can be the combination of several simple and specific supervised models, into a complete model.

2. Different configuration of the models, removing the ergodicity implemented in this modeling, and introducing structures that would be more suitable for describing the behavioral activity of the animal. An example of such models are the so-called left-to-right models, in which the underlying state sequence has the property that as time increases the state index increases (or at least stays the same), i.e. the states proceed from left to right. The combination of hierarchical modeling and different configuration may result in the model shown in figure 7.1. This figure shows a model combined of simple 2-4 state models embedded in a left-to-right model. The small models represent different activities in the experiment trial (see figure 3.1), and the left-to-right model represents the flow of activity in the experiment.

3. Introducing different parameters of the models. A short list of these parameters may include the following:

   (a) The temporal activity of the cells, i.e. the exact timing of action potentials, and not just the firing rates of cells.

   (b) The temporal cross-correlation functions between cells - in contrast with the firing rate cross-correlation functions used in this modeling.

   (c) Synchronous patterns of activity.

   (d) Different evaluation methods of firing rates of cells.

4. The isolation scores of cells may be utilized in a productive way in future work, in order to validate the results, and to allow information from highly contaminated cells to be used. These type of cells may still hold some valuable information for the modeling.

5. The dynamics of functional connectivities may be further investigated using the JPST tool, for the pairs of cells that show modification in their functional connectivity.

6. Comparing the model’s results with those achieved by the simulation of neural activity, i.e. an appropriate artificial neural network has to be constructed and its output should be modeled using HMM.
Pre Cue Modeling
Post Cue (Left)
Movement to the Left
Inter Trail Modeling
Movement to the Right
Post Cue (Right)

Figure 7.1: An example of HMM combined from simple task specific models, and the embedding left-to-right model.
Bibliography


BIBLIOGRAPHY


