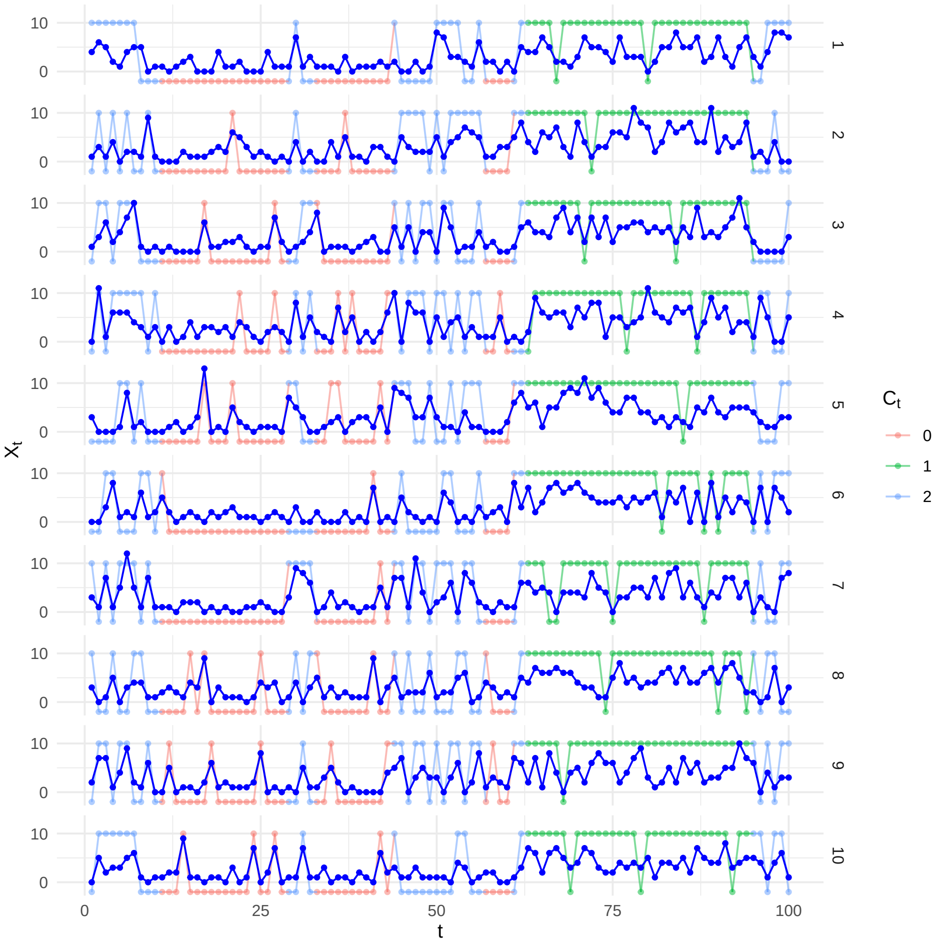
# A HIDDEN MARKOV MODEL OF VISUAL ATTENTION

NIELS RICHARD HANSEN

Exam project for the course *Models of Complex Systems*, 2021



Department of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, 2100 Copenhagen Ø, Denmark

*E-mail address*: `Niels.R.Hansen@math.ku.dk`.

## 1. Introduction

A neuron is the fundamental computational unit in the brain. It can receive input in the form of electro-chemical signals and it can emit output as well. The output is emitted as a *spike* – a rapid change in the membrane potential of the neuron. For the purpose of this project we can think of a spike as an instantaneous event. The only thing that then matters is the timing of spikes.

If we observe a neuron over a short time interval, say 50 ms, we can count the number of times it spikes as a quantification of how active it is. For a neuron in the visual cortex of the brain, we can use these numbers to tell how the neuron reacts when a subject is shown different visual stimuli.

If a neuron is *attending* to a particular stimuli, it will show a certain pattern of activity, and different stimuli may result in different patterns. If a test subject is shown two stimuli, a collection of neurons may then work in an uncoupled way with each attending to one of the stimuli independently, or they may work in a coupled way with most attending to the same stimuli.

The uncoupled attention is an example of *parallel processing*, while the coupled attention is an example of *serial processing.* We will in this project build a Bayesian network in the form of a Hidden Markov Model, which can capture both forms of processing.

An experiment will consist of the recording of spikes for $n$ neurons over a time period, e.g. 5 seconds. The recordings are then summarized as counts $X_{t,i} \in \mathbb{N}_0$ for $t = 1, \ldots, T$ and $i = 1, \ldots, n$, with $t$ indexing the time interval and $i$ the neuron. That is, $X_{t,i}$ is the number of spikes observed from time $50 \times (t-1)$ ms to time $50 \times t$ ms for neuron $i$. The data for a single experiment can thus be regarded as a $T \times n$ matrix

$$\mathbf{X} = \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} & \ldots & X_{1,n} \\ X_{2,1} & X_{2,2} & X_{2,3} & \ldots & X_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{T,1} & X_{T,2} & X_{T,3} & \ldots & X_{T,n} \end{pmatrix}.$$

Figure 1 shows one of the data sets available from the data file. There is a total of 10 data sets from 10 experiments available where there is a different number of neurons measured in each experiment. As figure 1 indicates for the data shown, the average activity is larger in the beginning an toward the end of the time period, while it is lower between interval 35 and 50.

With the Hidden Markov Model we will get a tool to explore data of this form and a systematic way to *decode* the attention mechanisms that the model expresses from the joint behavior of neurons.

## 2. Formalities

The project has three parts, and your report should reflect this by containing three corresponding sections. The report must be at most 15 pages including figures etc. (using a reasonable font and margins). The 15 pages may not contain code, and should be understandable without referring to code. However, the report must have an appendix with relevant code. The appendix does not count toward the 15 pages, but be careful to only include relevant code and do so in an organized way.

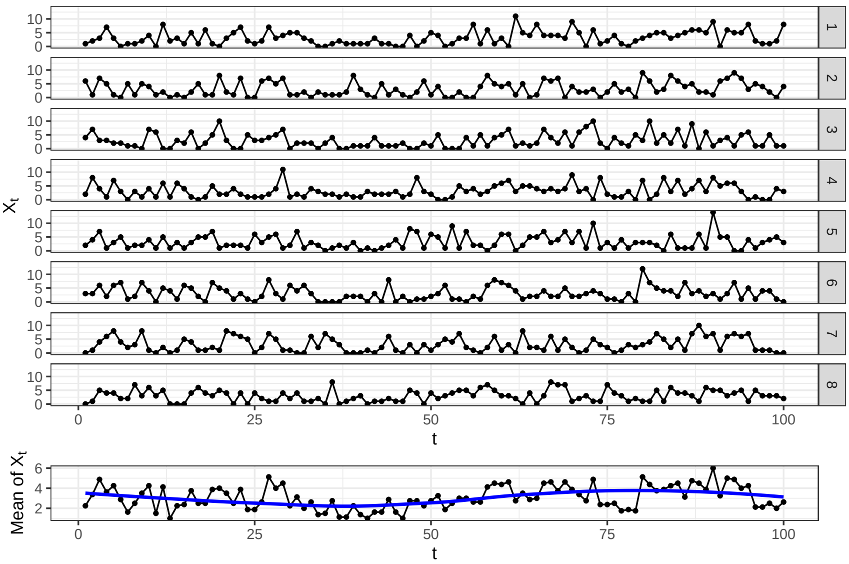


Figure 1. Data from one experiment with $T = 100$ time intervals and $n = 8$ neurons. The mean across the eight neurons is shown in the bottom figure together with a smoothed curve.

In Part I you are asked to implement simulation from a Bayesian network. The report must contain information on how you have implemented the simulation using a combination of clear text, mathematical symbol language and pseudo code. And it must contain examples of simulations using graphics and comments on the results you get.

In Part II of the project you are asked to implement inference of the unobserved nodes in the network. The report must contain explanations of the algorithms you have implemented and illustrations of the results obtained using the inference algorithms. The implementation should also be tested and the report must document the tests.

In Part III of the project you are asked to learn model parameters from data. This is a more open-ended part of the project. Some minimal questions are asked explicitly and one learning algorithm is suggested for further consideration. The report must present a solution to the minimal questions and some considerations about parameter learning using either simulated data or the data set provided.

The report is written by a group of 4–6 students and will form the basis for the oral exam of the course. The oral exam is individual, without preparation and takes 15 min. It consists of two parts:

- A short presentation (7 min) by the student, which is prepared in advance. The presentation is of *one* of the three parts of the report. Which part is entirely up to the individual student.
- Questions by the examiners (8 min) in the full report and curriculum of the course.

There are two different exam projects with different types of Bayesian networks. During the exam, questions will *not* be asked about specifics of the project you have not done, but generel questions in the curriculum related to the type of Bayesian network from the other project can be asked.

For the implementations you can use standard libraries but not libraries specifically written for graphical models. That is, it is *not* sufficient to present an implementation relying on some library for solving simulation, inference and learning problems by simply specifying the graphical model. You are welcome to use such libraries for testing, but it is not required.

Thus you must implement the algorithms needed using standard libraries. In Python, you may use NumPy, SciPy, scikit-learn, matplotlib, pandas etc.

## 3. A generative model

The generative model we will consider is a Hidden Markov Model with Bayesian network structure as illustrated in Figure 2. The variables, $C_1, \ldots, C_T \in \{0,1,2\}$ and the variables $Z_{1,1}, \ldots, Z_{1,n}, \ldots, Z_{T,1}, \ldots, Z_{T,n} \in \{0,1\}$ are all unobserved.

The CPDs for these variables are given by $P(C_1 = 2) = 1$,

$$P(C_{t+1} = d \mid C_t = c) = \gamma_{c,d}$$

for probability parameters $\gamma_{c,d} \in [0,1]$, and

$$P(Z_{t,i} = 1 \mid C_t = c) = \begin{cases} 1-\alpha & \text{if } c = 0 \\ \alpha & \text{if } c = 1 \\ 0.5 & \text{if } c = 2 \end{cases}$$

for a probability parameter $\alpha \in (0.5, 1)$. We can collect the $\gamma_{c,d}$ parameters into the matrix

$$\Gamma = \begin{pmatrix} \gamma_{0,0} & \gamma_{0,1} & \gamma_{0,2} \\ \gamma_{1,0} & \gamma_{1,1} & \gamma_{1,2} \\ \gamma_{2,0} & \gamma_{2,1} & \gamma_{2,2} \end{pmatrix},$$

whose row sums have to be 1.

The CPD $P(X_{t,i} \mid Z_{t,i} = z)$ is a Poisson distribution with mean value $\lambda_z > 0$, that is

$$P(X_{t,i} = x \mid Z_{t,i} = z) = e^{-\lambda_z} \frac{\lambda_z^x}{x!}.$$

for $x \in \mathbb{N}_0$ and $z = 0, 1$.

The variable $Z_{t,i}$ determines which of the two stimuli that neuron $i$ is attending at time $t$. Depending on the stimuli attended, the neuron will be more or less active as determined by the parameters $\lambda_0$ and $\lambda_1$. The interpretation of $C_t$ is that $C_t = 2$ represents parallel processing, where each of the stimuli is attended by the neuron with probability 0.5, while $C_t$ being in states 0 or 1 represent serial processing with most neurons either attending to stimuli 0 or 1, respectively.

You can throughout use the simplified parametrization of $\Gamma$ as

$$\Gamma = \begin{pmatrix} 1-\gamma & 0 & \gamma \\ 0 & 1-\gamma & \gamma \\ \beta/2 & \beta/2 & 1-\beta \end{pmatrix},$$

for $\gamma \in (0,1)$ and $\beta \in (0,1)$. The interpretation of this simplified parametrization is that the state variable $C_t$ can move from either of the two serial processing states to parallel processing with probability $\gamma$ and from parallel processing to one of the two serial processing states with probability $\beta$.

For part I of the project you need to:

- Implement forward simulation from the Hidden Markov Model. The implementation may be general, but must handle $n = 10$ and $T = 100$.

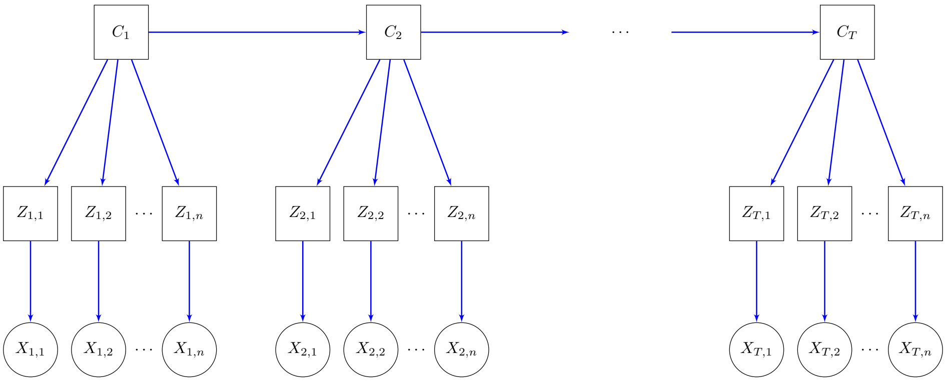


FIGURE 2. An illustration of the Bayesian network structure used in this project. The variables $X_{t,i}$ are the counts of observed spikes in the time intervals indexed by $t = 1, \ldots, T$ for the neurons $i = 1, \ldots, n$. The variables $Z_{t,i} \in \{0, 1\}$ are the attention variables indicating if neuron $i$ in time interval $t$ is attending stimuli 0 or 1. The variables $C_t \in \{0, 1, 2\}$ are state variables determining the probabilities of $Z_{t,i} = 1$ for all $n$ neurons in time interval $t$.

- Illustrate the implementation by generating example data and present them visually.
- Fit (multiclass) logistic regression model(s) of $C_t$ given $\mathbf{X}$ for one or more values of $t$ using (lots of) simulated data.

You can for the simulations use $\alpha = 0.9$, $\beta = 0.2$, $\gamma = 0.1$, $\lambda_0 = 1$ and $\lambda_1 = 5$. But you are also welcome to test what happens for other choices of parameters.

## 4. Inference of hidden nodes

Only the spike counts, $X_{t,i}$, are in practice observed, and prediction of the unobserved attention variables, $Z_{t,i}$, and state variables, $C_t$, is an inference problem.

In part II of the project you need to:

- Implement inference algorithms for computing the conditional distribution of each of the variables $C_1, \ldots, C_T$ and $Z_{1,1}, \ldots, Z_{1,n}, \ldots, Z_{T,1}, \ldots, Z_{T,n}$ given $\mathbf{X}$.
- Test the inference algorithms using simulated data.
- Apply the inference algorithm on the data in the data file and present the results.

Note that for inference the $X_{t,i}$-variables are conditioned upon and they thus act as evidence. The unobserved variables are discrete taking values in $\{0, 1\}$ and $\{0, 1, 2\}$. Message passing can therefore be implemented via arrays, and only $2 \times 3$ and $3 \times 3$ 2d-arrays are, in fact, needed.

For very small values of $n$ and $T$, e.g. $n = 2$ and $T = 5$, it is possible to brute-force compute the full conditional distribution for testing. However, it may be more convenient to test the implementation in other ways. You may, for instance, observe that

$$1(Z_{t,i} = z) - P(Z_{t,i} = z \mid \mathbf{X} = \mathbf{x})$$

has mean 0 and likewise for $C_t$. Using simulations you can compute such quantities, with $P(Z_{t,i} = z \mid \mathbf{X} = \mathbf{x})$ computed by the inference algorithm, and empirically check if their averages across many replications of the simulations are zero.

Other possibilities for testing your implementation includes checking the calibration identities of clique beliefs. And you can also test the implementation by comparing results to the results from the logistic regression models found using forward simulation in Part I. However, note that the logistic regression models will only be approximations of the true conditional distributions of $C_t \mid \mathbf{X}$.

## 5. Learning of the parameters

The objective of this part is to learn the parameters $\alpha, \beta, \gamma, \lambda_0, \lambda_1$ from data. The ultimate goal is to learn the parameters from observing only $\mathbf{X}$, but the problem is broken down so you first consider learning from a complete observation of all variables.

- Suppose first that all variables, $C_1, \ldots, C_T$, $\mathbf{Z}$ and $\mathbf{X}$ are observed. Implement learning of the parameters as follows:
  - compute $\hat{\lambda}_0$ as the average of the $X_{t,i}$-s for which $Z_{t,i} = 0$ and $\hat{\lambda}_1$ as the average of the $X_{t,i}$-s for which $Z_{t,i} = 1$
  - compute $\hat{\alpha}$ as the relative frequency of the events $Z_{t,i} = C_t = 0$ and $Z_{t,i} = C_t = 1$
  - compute $\hat{\beta}$ as the relative frequency of transitions from $C_t = 2$ to $C_{t+1} \in \{0, 1\}$ and $\hat{\gamma}$ as the relative frequency of transitions from $C_t \in \{0, 1\}$ to $C_{t+1} = 2$.

  What you effectively need for the two latter points above is to tabulate the values of $(Z_{t,i}, C_t)$ and $(C_t, C_{t+1})$.
- Test the implementation using simulated data.
- Proceed to implement learning with only $\mathbf{X}$ observed. One simple solution is the hard-assignment EM algorithm, which combines the inference algorithm from Part II with the learning from complete observations. Thus iteratively compute

$$\hat{Z}_{t,i} = \arg\max_z P(Z_{t,i} = z \mid \mathbf{X} = \mathbf{x})$$
$$\hat{C}_t = \arg\max_c P(C_t = c \mid \mathbf{X} = \mathbf{x}),$$

  and update the parameters using $\hat{C}_1, \ldots, \hat{C}_T$, $\hat{\mathbf{Z}}$ and $\mathbf{X}$ as observations. Explore convergence using simulated data and the data from the data file.

Alternatives to the hard-assignment EM algorithm are gradient ascent and the soft-assignment EM algorithm. You are welcome to explore such alternative algorithms, but this is not required.

## 6. Data

Data for this project comes in the file `proj_HMM.zip`, which is a zip-file. It contains 10 files, each file containing a data table with one column called `t` and the other columns called `X1`, `X2` etc. Each data table corresponds to one experiment and the `X`-columns correspond to the $\mathbf{X}$ data matrix.

## 7. Postscript

The cover page shows a simulation from the HMM with the blue points and lines showing the values of the $X_{t,i}$-variables, with the colored points and lines in the background indicating values of the $Z_{t,i}$-variables (high is 1 and low is 0) and with the colors indicating the values of the $C_t$-variables. Note how the color is the same across neurons as all neurons share the same value of $C_t$ in a given time interval, while the values of $Z_{t,i}$ are unique to the neuron, though somewhat similar across neurons when $C_t = 0$ or $C_t = 1$.

The model is based on the paper: Distinguishing between parallel and serial processing in visual attention from neurobiological data, 2020, *R. Soc. open sci. 7:191553*, by Li K, Kadohisa M, Kusunoki M, Duncan J, Bundesen C, Ditlevsen S.