Homework 12 — Decoding the Monkey's Paw

36-467/667 (Fall 2020)

Due at 6 pm on Thursday, 3 December 2020

Nerve cells (or "neurons") communicate and process information by transmitting little electrical impulses to each other, called "spikes". Many neurons use "rate codes", where the number of spikes they produce in a short period of time encodes information either about some aspect of the world the organism is sensing, or about how the organism is acting or is going to act.

For example, when very small electrodes are inserted into certain motor-control regions of the brains of monkeys, so that neuroscientists can record from individual neurons, some cells are found to encode the direction in which the monkey intends to move its hand. Specifically, a neuron has a preferred direction vector $\vec{b}$, and the when the monkey intends to move its hand with velocity $\vec{v}$, the average number of spikes over a short interval is $a + \vec{b} \cdot \vec{v}$, plus or minus some amount of noise. A neuron which behaves like this is said to show "directional tuning", and $\vec{b}$ is its "preferred direction"[1].

A common model is to divide time into small "bins", of duration h, and say that during time $[t, t+h)$, the number of spikes produced by neuron i, $X_i(t)$, is

$$X_i(t) \sim \text{Poisson}\left(e^{a_i + \vec{b}_i \cdot \vec{v}(t)}\right) \tag{1}$$

with the different neurons being independent given $\vec{v}(t)$. This is completed by a model for how the velocity changes,

$$\vec{v}(t+h) \sim \mathcal{N}(\vec{v}(t), \sigma^2 \mathbf{I}) \tag{2}$$

where $\mathbf{I}$ is the identity matrix.

In "decoding", we observe the spiking activity of neurons, and try to infer the intended velocity of the hand[2].

In this week's data set, we have recordings of multiple motor-cortex neurons made while a monkey was controlling a cursor on a screen. The task the monkey was being trained to do was to move the cursor to an on-screen target, so the data is divided into "reaches", of which there were 419. The duration of each time bin was 10 milliseconds, and the data file records the number of spikes for each neuron during each time bin. Velocity was two dimensional, and measured in centimeters per second for each time bin.

1. (10) Think of the model in equations (1) and (2) as a hidden-Markov or state-space model. What are the state variables? What is the dimension of the state? Is it discrete or continuous? What are the observable variables? What is the dimension of the observables? Are the observables discrete or continuous (or a mix of both)?

2. (5) Neurons where the firing rate (expected number of spikes per unit time) follows $a + \vec{b} \cdot \vec{v}(t)$ are said to have "cosine tuning curves". *Without* looking up that phrase, explain the "cosine" part of the name. *Hint*: what's another way to write the dot product?

3. (10) Equation (2) is often said to represent a "smoothness constraint". Explain why.

4. (15) Using the first 410 reaches in the data, estimate the intercept a_i and preferred direction $\vec{b}_i$ for each neuron by running Poisson regressions. You should get 147 estimated parameters. Make a

[1] For more on such models of neural coding, see, for example, Dayan and Abbott (2001).

[2] In addition to its scientific interest, good decoding techniques make it possible to develop prosthetic limbs which people can control by thinking, like natural limbs.

plot of the estimated preferred direction vectors. *Hints:* (1) The `glm` function will run Poisson regressions, with a syntax similar to `lm`; (2) write a loop to do all the estimates, *or* split the data by neuron; (3) you could show all the estimated $\vec{b}_i$s in one figure that would look like a starburst.

5. (9) Using the first 410 reaches in the data, estimate σ^2.

6. In these problems, display the code. *General hint*: look at the example of setting up and using a particle filter given in the slides.

   a. (10) Modify the particle filtering code given in the slides to handle a model where the state variable is two-dimensional, and the observations are vector-valued. *Hint*: the easiest way to do this is to use a three-dimensional array for the states, and a two-dimensional array for the observations.

   b. (5) Write a function which will move a particle at random, with the correct distribution.

   c. (5) Write a function which will give the conditional probability of an observation *vector* given the state.

7. (15) Using the parameters you estimated in problems (4) and (5), and the code you wrote in (6), use the particle filter to estimate the two-dimensional velocity time series during each of the last 9 reaches; use at least 30 particles. (Your code should be fast enough that using many more is feasible.) For each of those last 9 reaches reach, make a plot showing (i) the true velocity trajectory, (ii) the mean of the particle filter, and (iii) the distribution of particles. Use a third dimension or color to indicate time within each reach. You should have 9 plots. Comment on the figures.

8. (5) Explain why I asked you to use distinct parts of the data to estimate the parameters and to run the particle filter?

9. (1) How much time did you spend on this problem set?

Presentation Rubric (10): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. All plots, tables, etc., are generated automatically by code embedded in the R Markdown file. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All quantitative and mathematical claims are supported by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. Code is properly integrated with a tool like R Markdown or knitr, and both the knitted file and the source file are submitted. The code is indented, commented, and uses meaningful names. All code is relevant, without dangling or useless commands. All parts of all problems are answered with coherent sentences, and raw computer code or output are only shown when explicitly asked for.

References

Dayan, Peter, and Laurence F. Abbott. 2001. *Theoretical Neuroscience*. Cambridge, Massachusetts: MIT Press.