Neural Decoding

Computational Neuroscience by University of Washington

This exercise is based on a set of artificial "experiments" that we've run on four simulated neurons that emulate the behavior found in the cercal organs of a cricket. Please note that all the supplied data is synthetic. Any resemblance to a real cricket is purely coincidental.

In the first set of experiments, we probed each neuron with a range of air velocity stimuli of uniform intensity and differing direction. We recorded the firing rate of each of the neurons in response to each of the stimulus values. Each of these recordings lasted 10 seconds and we repeated this process 100 times for each neuron-stimulus combination.

We've supplied you with a .mat file for each of the neurons that contains the recorded firing rates (in Hz). These are named $neuron1$, $neuron2$, $neuron3$, and $neuron4$. The stimulus, that is, the direction of the air velocity, is in the vector named $stim$.
Download the file:

[tuning.mat]

and save it into your MATLAB/Octave directory. To load the data, use the following command:

load('tuning.mat')

The equivalent data files for Python 2.7 and Python 3.4 are:

[tuning_2.7.pickle]

[tuning_3.4.pickle]

To load the data, make sure you are in the same directory you saved it and add the following (shown for 2.7) to your script:

```
import pickle

with open('tuning_2.7.pickle', 'rb') as f:
    data = pickle.load(f)

```

This will load everything into a dict called data, and you'll be able to access the stim and neuron responses using data['stim'], data['neuron1'], etc. (In general, data.keys() will show you all the keys available in the dict.)

The matrices contain the results of running a set of experiments in which we probed the synthetic neuron with the stimuli in stim. Each column of a neuron matrix contains the firing rate of that neuron (in Hz) in response to the corresponding stimulus value in $stim$. That is, $n\mathrm{th}$ column of $neuron1$ contains the 100 trials in which we applied the stimulus of value stim(n) to $neuron1$.

Part A

Plot the tuning curve-- the mean firing rate of the neuron as a function of the stimulus-- for each of the neurons.

Part B

We have reason to suspect that one of the neurons is not like the others. Three of the neurons are Poisson neurons (they are accurately modeling using a Poisson process), but we believe that the remaining one might not be.

Which of the neurons (if any) is NOT Poisson?

Hint: Think carefully about what it means for a neuron to be Poisson. You may find it useful to review the last lecture of week 2. Note that we give you the *firing rate* of each of the neurons, not the spike count. You may find it useful to convert the firing rates to spike counts in order to test for "Poisson-ness", however this is not necessary.

In order to realize why this might be helpful, consider the fact that, for a constant a and a random variable X, $\mathrm{E}[aX]=a\mathrm{E}[X]$ but $Var(aX)=a^2Var(X)$. What might this imply about the Poisson statistics (like the Fano factor) when we convert the spike counts (the raw output of the Poisson spike generator) into a firing rate (what we gave you)?

Part C

Finally, we ran an additional set of experiments in which we exposed each of the neurons to a single stimulus of unknown direction for 10 trials of 10 seconds each. We have placed the results of this experiment in the following file:

pop_coding.mat

You should save the file into your MATLAB/Octave directory and import the data using the following command:

load('pop_coding.mat')

The equivalent python files are:

pop_coding_2.7.pickle

pop_coding_3.4.pickle

These can be loaded in the same way as described in question 7 above.

pop_coding contains four vectors named $r1$, $r2$, $r3$, and $r4$ that contain the responses (firing rate in Hz) of the four neurons to this mystery stimulus. It also contains four vectors named $c1$, $c2$, $c3$, and $c4$. These are the basis vectors corresponding to neuron 1, neuron 2, neuron 3, and neuron 4.

Decode the neural responses and recover the mystery stimulus vector by computing the population vector for these neurons. You should use the maximum average firing rate (over any of the stimulus values in 'tuning.mat') for a neuron as the value of r_{max} for that neuron. That is, r_{max} should be the maximum value in the tuning curve for that neuron.

What is the direction, in degrees, of the population vector? You should round your answer to the nearest degree. Your answer should contain the value only (no units!) and should be between 0° and 360°. If your calculations give a negative number or a number greater than or equal to 360, convert it to a number in the proper range (you may use the mod function to do this).

You may need to convert your resulting vector from Cartesian coordinates to polar coordinates to find the angle. You may use the atan() function in MATLAB to do this. Note that the the convention we're using defines 0° to point in the direction of the positive y-axis and 90° to point in the direction of the positive x-axis (i.e., 0 degrees is north, 90 degrees is east).