

University: Sharif University of Technology

Department: Electrical Engineering

Course Name: Advanced Neuroscience

Homework 2 Report

Studying the Population Response Structure

Student Name: Ali Shahbazi

Student ID: 98101866

Instructor: Prof. Ali Ghazizadeh

Academic Semester: 2023 Spring

Step 1

- a) Here we plot PSTH of 6 random units:

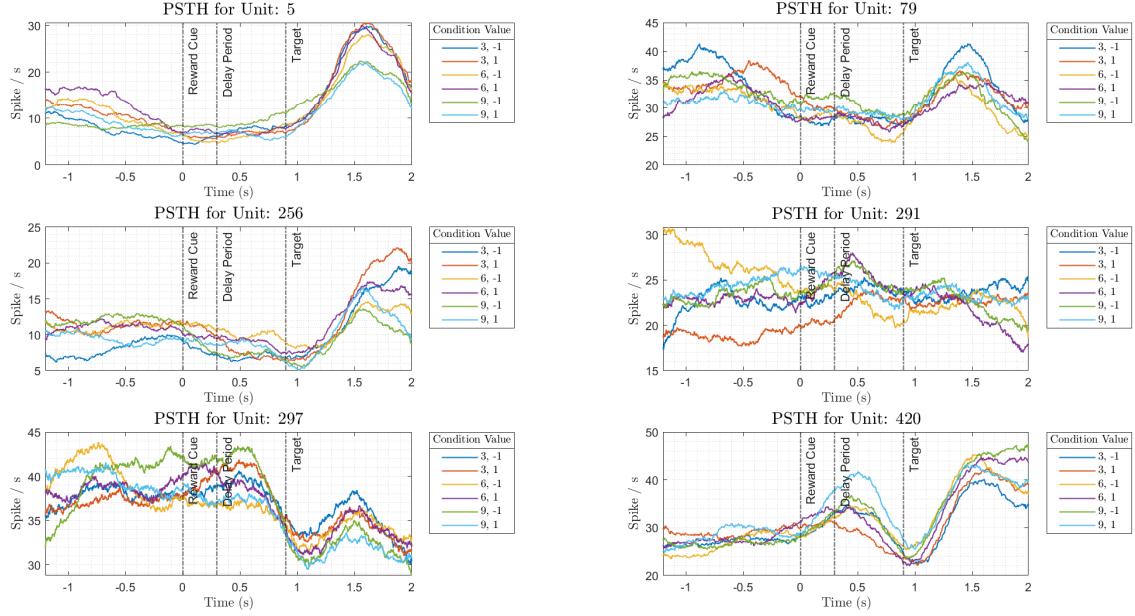


Figure 1: PSTH of different units

As we can see in [Figure 1](#), it seems that different neurons respond differently to the cue. In addition, we can see different base firing rates for different neurons which means they are not likely to be similar.

- b) In [Figure 2](#), the average PSTH of all units is plotted for different conditions (150ms time window). Hence, we can see that after the *target* is presented, firing rate increases which means they are target response neurons.

We can observe that PSTH of conditions are pair-wised, conditions with same reward expected value are close to each other. It means that these neurons are coding the reward expected value far better than cue location. Also we can see that neurons are more active when they feel a low reward expected value, rather than high reward expected value (blue-red rate is higher than green-cyan).

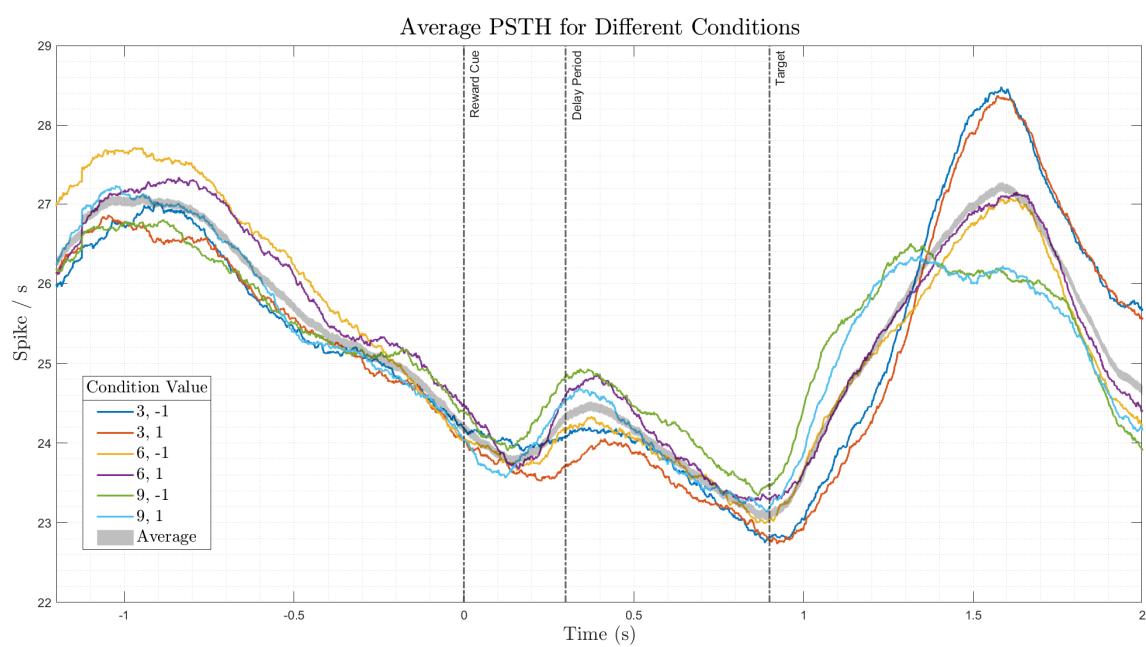


Figure 2: Average PSTH of all units

Step 2

If we get average firing rate on trials for a given condition and unit, then calculate spike count in 100ms time bins, we can write a linear model explaining the spike count:

$$\mathbf{y}(t) = \beta_0(t) + \beta_1(t)\mathbf{x}_1(t) + \beta_2(t)\mathbf{x}_2(t) + \beta_3(t)\mathbf{x}_3(t) + \beta_4(t)\mathbf{x}_4(t) + \beta_5(t)\mathbf{x}_5(t)$$

In which $\mathbf{y}(t), \mathbf{x}_i(t) \in \mathbf{R}^{192 \times 1}$ for a given t ; $\mathbf{x}(t)$ is a vector containing only 0 everywhere except the trials where that condition is present for that unit (filled with 1). We used *fitglm()* function to regress neural responses against these variables. Values of β_i are derived and p-values of β_i are calculated. Also, the model p-value has been derived.

[Figure 3](#) shows p-values for β_i across time. In the last row, model p-value is shown. Each p-value shows the probability of that β_i to be zero. Hence, the lower the p-value, the higher the neuron is coding the average PSTH in that time bin.

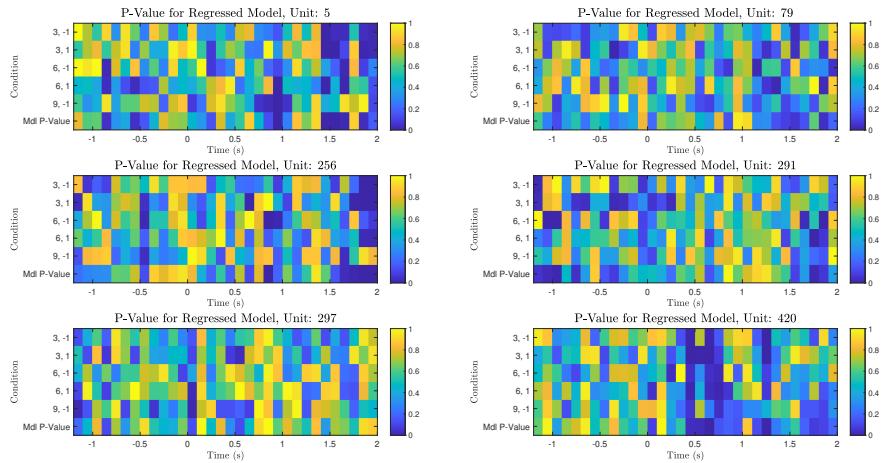


Figure 3: P-Value of different units across time

In [Figure 4](#) you can see model p-value of all units in time. Blue colors are indicating lower p-values. And in [Figure 5](#), p-values are cut by a threshold of 0.01, which means yellow bars have p-value less than 0.01 and blue bars are above threshold.

If we plot the histogram of p-values in [Figure 4](#), we get [Figure 6](#). This figure shows that the visual difference we explained in step 1, is generated by this peak of p-values.

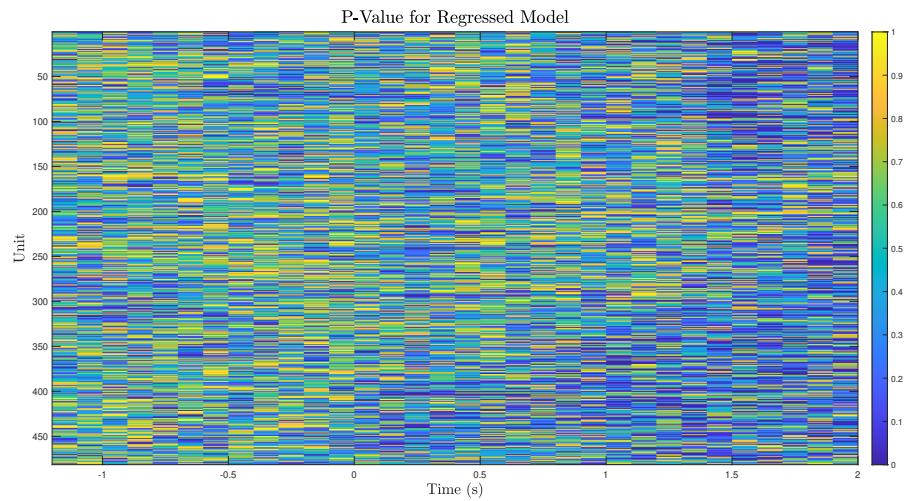


Figure 4: P-Value of all units in time

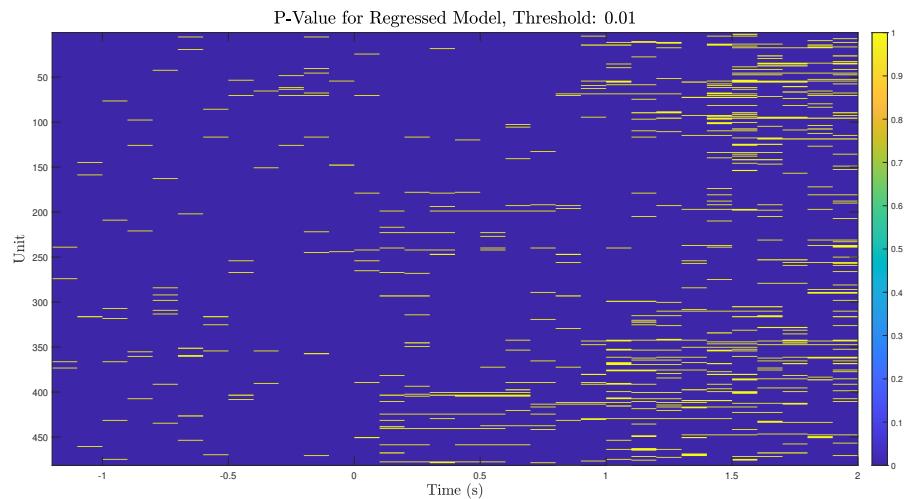


Figure 5: P-Value of all units with threshold (0.01)

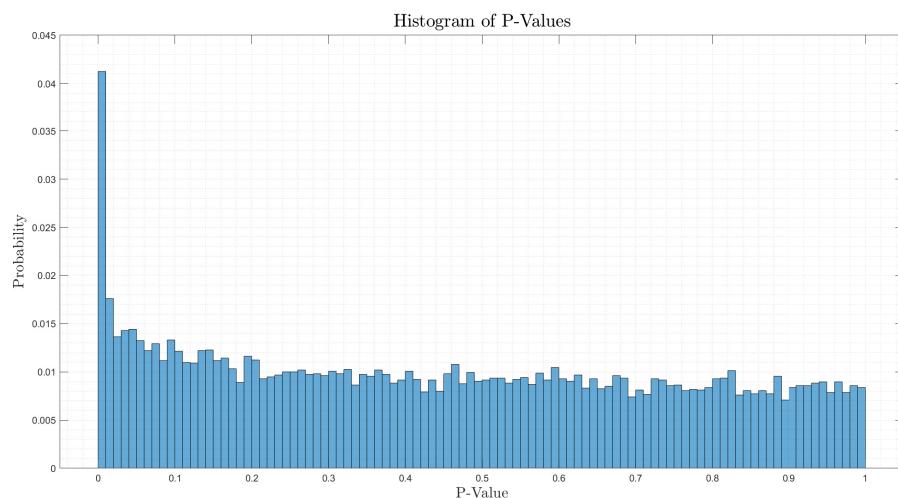


Figure 6: Histogram of p-values

But if we don't use time bins and just regress the conditions to all-time spike count (eliminating time), we can derive a single p-value for each unit in each condition. In that case [Figure 7](#) is obtained. If we set a threshold 0.01 for p-values we obtain [Figure 8](#). Here we can see units which significantly encode the task conditions.

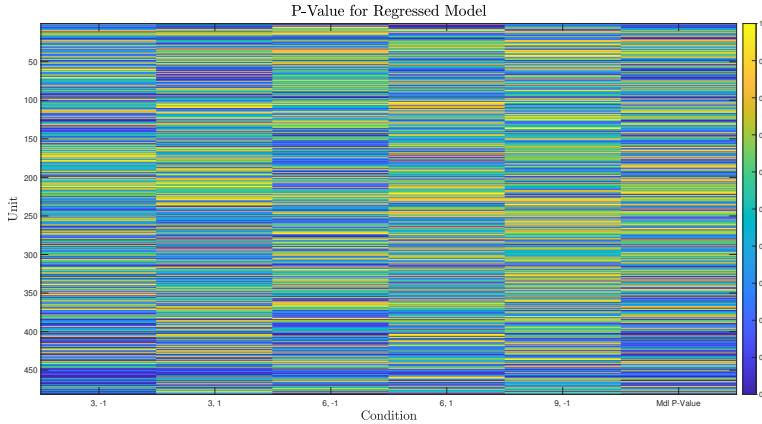


Figure 7: P-Value of units over conditions

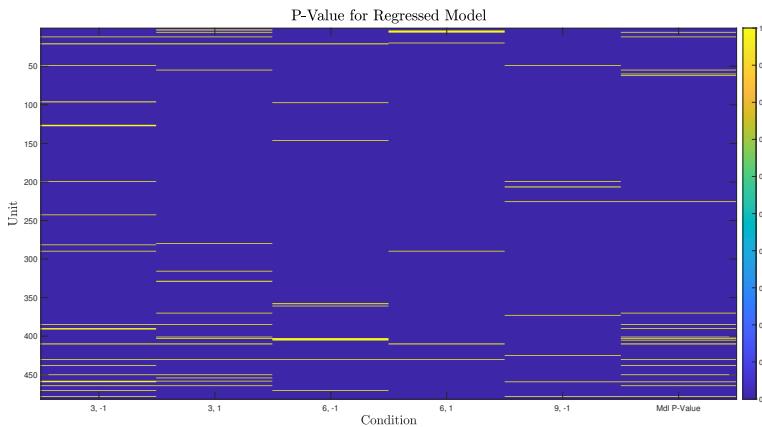


Figure 8: P-Value of units over condition with threshold (0.01)

Some of the unit numbers are reported here:

Units for condition 1:

12	21	49	96	126	127	199	242	281	289	384	389	390	409	429	437	449	457
458	463	469	477														

Units for condition 2:

3	6	12	21	55	279	315	328	369	384	400	402	409	429	449	453	457	463
---	---	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Units for condition 3:

21	97	146	357	360	402	403	404	429	469								
----	----	-----	-----	-----	-----	-----	-----	-----	-----	--	--	--	--	--	--	--	--

Units for condition 4:

4	5	6	20	289	409	429											
---	---	---	----	-----	-----	-----	--	--	--	--	--	--	--	--	--	--	--

Units for condition 5:

49	199	206	225	372	424	458	477										
----	-----	-----	-----	-----	-----	-----	-----	--	--	--	--	--	--	--	--	--	--

Step 3

For a given unit and condition, we have a vector containing the firing rate of that neuron in that specific condition. Since we have 6 different conditions we can make 6 matrix of size $N \times T$; which N is the number of units and T is the time. To get 3 same principal component (PC) for all conditions, we concatenate these 6 matrix to a $N \times 6T$ matrix and run PCA¹ on the units to get 3 basic PCs of these N neurons. We get just 3 PCs because we want to plot the results but if you check the *latent* output of function `pca()`, you will get a number for each PC which indicates how much variance of data that PC is presenting.

After running PCA, we de-concatenate those 6 matrix in time and plot [Figure 9](#). Each trajectory starts from a square and ends at a circle. As we can see, all 6 conditions start close to each other but from a time later, they divide into three groups of trajectories. As expected, reward expected trajectories differ while cue location does not have strong effect on the trajectory.

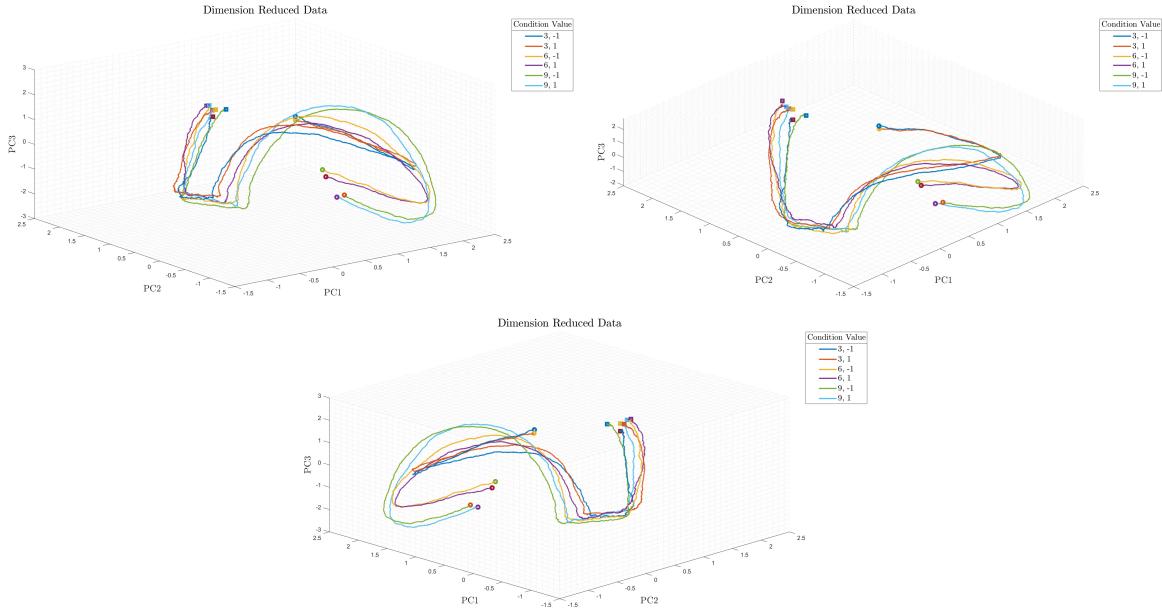


Figure 9: Trajectories of 6 condition in first 3 PCs (not shuffled)

¹Principal Component Analysis

Step 4

In this section we use CFR² discussed in the paper. Since the run time of the code for high number of surrogates is too long, I set `numSurrogates = 3`. For this number of shuffling we don't expect so much difference in trajectories but they should not be easily discriminated by reward expected value anymore. [Figure 10](#) shows the shuffled trajectories.

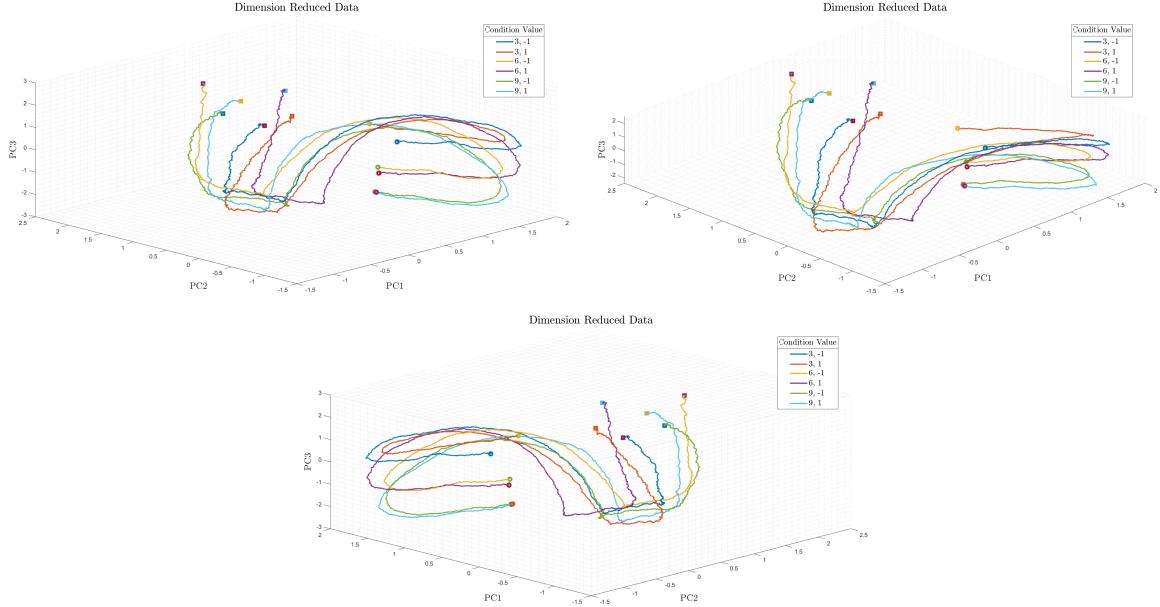


Figure 10: Trajectories of 6 condition in first 3 PCs (shuffled)

As a conclusion, there is a population coding which teaches us more than what is expected from single unit analysis because shuffled data does not convey enough information about conditions.

²Corrected Fisher Randomization