Sofia Lima

15-686 Neural Computation

Dr. Tai-Sing Lee

31 March, 2023

Problem Set 4: Bayesian decoding of neural activity

**Overview of assignment** In this assignment you will take data recorded from an actual experiment, courtesy of David Redish's lab, and decode the neural data to determine the behavioral variable being represented, in our case, locations in the environment.

First, unzip the data and code file from the assignment folder, and load('hippocampus data.mat').

You will be provided with the following data structures:

- **spike matrix**; spike matrix is an NxT matrix, where N is the number of simultaneously recorded neurons and T is the number of time bins throughout the experiment. Each index (entry) in the matrix contains the number of spikes a particular neuron fired in a particular time window. The recording session was approximately 40 minutes long and spikes were binned into 10 ms time windows, so T is on the order of 40*60*100.
- **pos**; pos is a Tx2 matrix describing the animal's location in the environment during each time window. The first column is x location and the second is y location. This is called the tracking data and each row of this matrix is referred to as a position sample. Do a quick plot to see what the animal's behavior was like during this experiment: plot(pos(:,1),pos(:,2),.)
- **behavior time**; behavior time is a 2x2 matrix, with each row containing the start and stop time bins for decoding the animal's trajectory in Part 2.
- **replay time**; replay time is 3x2 matrix, with each row containing the start and stop time bins for decoding replay events in Part 3.

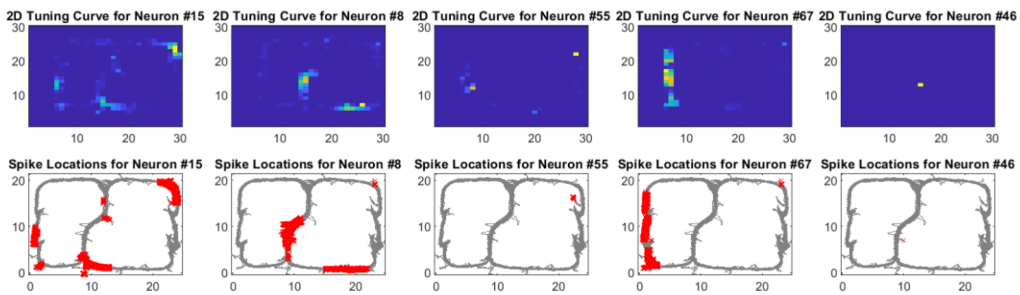
```
plot(pos(:,1),pos(:,2),.)
```

**Part 1: Plotting tuning curves** (2 pts)

In this part of the assignment you should learn to see how tuning curves (place fields in our application) for each cell is constructed. Your job then is simply to plot out the tuning curves (surfaces) in two different ways, and to study the cells' tuning curves.

*Report*: plot tuning curve of 5 neurons; 2D tuning curves of 4 cells including #8, location of animal in spike (red) and all tracking data (grey)

Shown below are the 2D tuning curves at location of spike (red Xs) for 5 neurons from this dataset. Neuron #15 (provided default), Neuron #8, and three randomly selected neurons are shown. I repeated this random selection until I found "especially nice place fields". Here, I aim to explain different patterns exhibited in the data.




Interesting behaviors are shown in these neurons. Neuron number 8 has strong responses to certain stretches of locations. Neuron #55 has strong but very specific responses to single points of locations (seemingly near feeding locations). Neuron #46 on the far right shows an extreme case. Neuron #67 has strong responses to a particular wall as well as some reference corner.

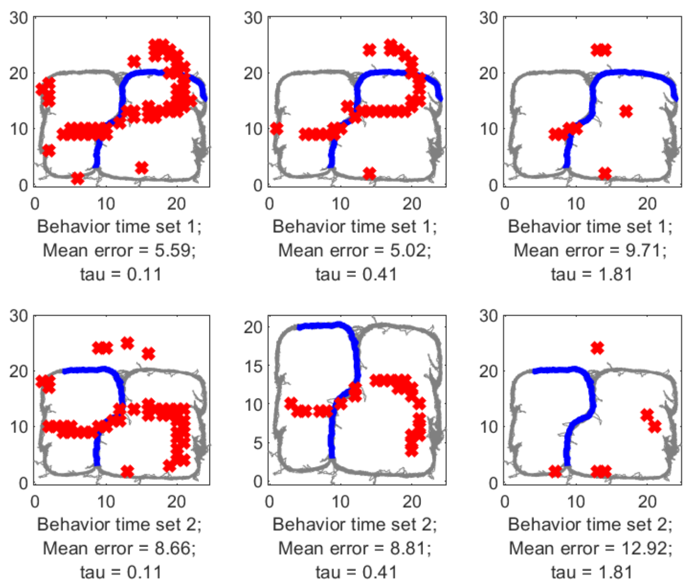
**Part 2: Decode paths taken by the animal during behavior** (5 pts)

```
function [decoded locations]=bayesDecode(spike matrix,tuning matrix,behavior time indiv,τ)
```

Bayes Rule postulates that the posterior probability of the spike count vector given a location in environment $P(n|x)$ is proportional to the likelihood of each location in the environment given the spike count $P(x|n)$. Assume the data follows a Poisson distribution. $P(n|x) = \prod_{i=1}^{N} P(n_i|x) = \prod_{i=1}^{N} \frac{\tau f_i(x)}{n_i!} e^{-\tau f_i(x)}$ where $n_i$ is the number of spikes observed from neuron $i$ over time window $\tau$ and $f_i(x)$ is the average firing rate of neuron $i$ at location $x$. Calculate mse.

*Report*: bayesDecode routine; for 3 runs plot tracking data (grey), actual path during decoding time window(blue), and decoded locations (red X) (with mse and τ underneath); (each with 2 periods of time to decode over = 6 plots)

Shown below are the decoded paths in red using 3 different time windows for 2 behavior periods where the animal was in the blue locations.




1. increase amount of data for decoding by increasing τ. Explain tradeoff (2 pt)

   Shown below are the decoding error results from 10 settings for tau for 2 sets of behavior time points.

   For behavior_time set #1 and tau = 0.11, the mean error is 5.59
   For behavior_time set #1 and tau = 0.21, the mean error is 5.07
   For behavior_time set #1 and tau = 0.31, the mean error is 4.93
   For behavior_time set #1 and tau = 0.41, the mean error is 5.02
   For behavior_time set #1 and tau = 0.51, the mean error is 5.37
   For behavior_time set #1 and tau = 0.81, the mean error is 6.57
   For behavior_time set #1 and tau = 1.81, the mean error is 9.71
   For behavior_time set #1 and tau = 2.81, the mean error is 9.62
   For behavior_time set #1 and tau = 5.81, the mean error is 8.33
   For behavior_time set #1 and tau = 10.81, the mean error is 20.36

For behavior_time set #2 and tau = 0.11, the mean error is 8.66
For behavior_time set #2 and tau = 0.21, the mean error is 8.77
For behavior_time set #2 and tau = 0.31, the mean error is 8.89
For behavior_time set #2 and tau = 0.41, the mean error is 8.81
For behavior_time set #2 and tau = 0.51, the mean error is 8.88
For behavior_time set #2 and tau = 0.81, the mean error is 9.75
For behavior_time set #2 and tau = 1.81, the mean error is 12.92
For behavior_time set #2 and tau = 2.81, the mean error is 10.28
For behavior_time set #2 and tau = 5.81, the mean error is 11.99
For behavior_time set #2 and tau = 10.81, the mean error is 20.50

These values and the images above demonstrate that the window size tau, and hence the number of windows, impacts the decoding performance.  Fewer number of windows means you can only decode fewer (more general) locations. In other words, fewer windows each cover more area that has to be decoded in each step, which will surely impact precision and accuracy.

On the other hand, a small tau means that many neurons are being tuned to different locations, and it could be hard to tell where the rat is.

Mathematically, a large tau decreases the maximum likelihood during the bayes decoding, since it is negated in the exponent of the second term. A smaller maximum likely location reduces the confidence for decoding.

2. Now use P(x) deduced from the behavioral data to do the decoding. Improves accuracy? When would you want to incorporate P(x) in decoding rat's position.

   Shown below are the decoding error results from 10 settings for tau for 2 sets of behavior time points. Here, the decoding function computes the posterior as a function of the prior p(x).

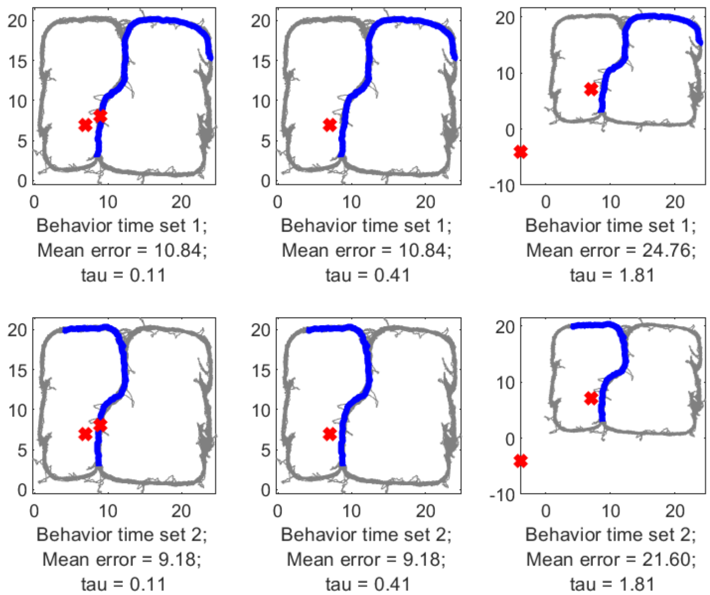
For behavior_time set #1 and tau = 0.11, the mean error is 10.84
For behavior_time set #1 and tau = 0.21, the mean error is 10.84
For behavior_time set #1 and tau = 0.31, the mean error is 10.84
For behavior_time set #1 and tau = 0.41, the mean error is 10.84
For behavior_time set #1 and tau = 0.51, the mean error is 10.84
For behavior_time set #1 and tau = 0.81, the mean error is 11.39
For behavior_time set #1 and tau = 1.81, the mean error is 24.76
For behavior_time set #1 and tau = 2.81, the mean error is 25.92
For behavior_time set #1 and tau = 5.81, the mean error is 25.92
For behavior_time set #1 and tau = 10.81, the mean error is 25.92

For behavior_time set #2 and tau = 0.11, the mean error is 9.18
For behavior_time set #2 and tau = 0.21, the mean error is 9.18
For behavior_time set #2 and tau = 0.31, the mean error is 9.18
For behavior_time set #2 and tau = 0.41, the mean error is 9.18
For behavior_time set #2 and tau = 0.51, the mean error is 9.18
For behavior_time set #2 and tau = 0.81, the mean error is 10.47
For behavior_time set #2 and tau = 1.81, the mean error is 21.60
For behavior_time set #2 and tau = 2.81, the mean error is 23.63
For behavior_time set #2 and tau = 5.81, the mean error is 23.63
For behavior_time set #2 and tau = 10.81, the mean error is 23.63

The accuracy here is higher compared to question 1 without including the prior term.

You would want to incorporate P(x) in decoding rat's position when you have high confidence in the prior belief in the distribution of spatial locations. This could be achieved by having a lot of data for each location.

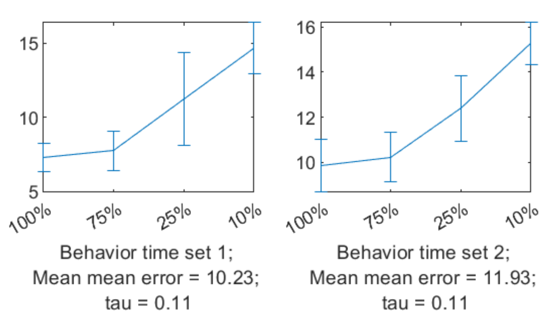
Shown below are the decoded locations in red during 2 sets of behavior time periods and the ground truth trajectory in blue. The pattern across window sizes is similar to question 1 such that the error is higher with larger values tau. We see higher errors overall compared to question 1.




3. How would dropout adversely affect the performance of decoder? Plot decoding errors w/ 100%, 75%, 25%, 10% of neuron samples used (try 10 or more random subsets and get average error).

Shown below are the decoding errors from various amounts of dropout (note that x labels indicated amount of data kept for decoding). The lines and error bars represent the mean and standard deviation from 10 runs of each dropout scenario.

As more data is dropped out, error increases. Dropout is a useful method for quantifying uncertainty.

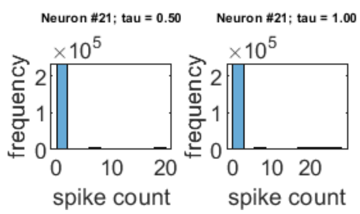


**Part 3: Models of Neuronal Variability** (2 pts)

How true is assumption that neuronal variability follows Poisson? Design an analysis to assess whether spike counts at each location follow Poisson (try τ=0.5 and 1).

Shown below are the histograms of spike counts for a particular neuron for a particular location. Many possible combinations of neurons and locations have 0 signal, which is a difficult histogram to analyze. To ensure plotting a neuron with some signal, I selected the neuron with the max value in spike_matrix, and the location with the max spike count observed.

These distributions seem to follow a Poisson distribution such that the domain is from 0 to positive infinity, and there seems a small mean with small variance.




This suggests that rather than an incorrect assumption about the distribution of this data, the error in our decoding in part 2 could be the result of the size of the dataset, or the naïve assumption that all locations are independent of each other as given by the product (which becomes a sum when you take the log) of the probabilities.

To generate these plots, I picked the two values of tau specified on the assignment. I iterated through all timepoints in the spike train for the selected neuron, and summed the spike counts for the window centered at each timepoint. I stored all the spike counts for each binned location for each binned time step. I then selected the location that had the largest spike count, and used the histogram plotting function in matlab on this set of spike counts for this particular location for this particular neuron.

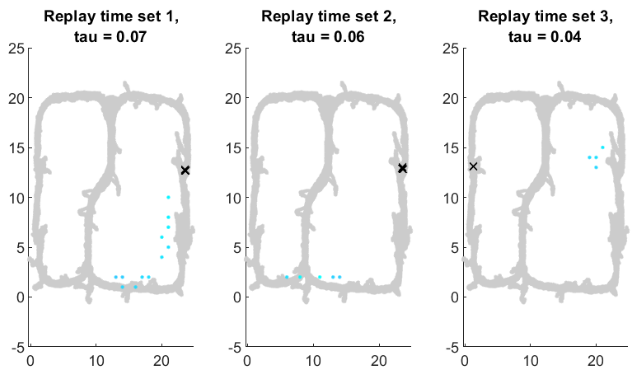
**Part 4: Decode paths expressed mentally, while the animal is paused in the environment** (2 pts)

`plotTrajectory.m`

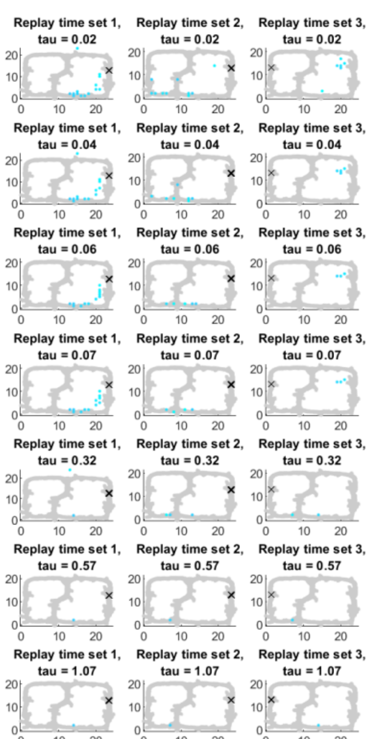
Input rows of replay_time. Replays occur on faster timescale than behavior → smaller τ (try a few)

*Report*: single plot for each replay event; (label number/row and τ); describe observations of different τ

Shown below are the plots for each replay event with the “nicest” results such that the dots form a path.




Below are the 7 values of tau I tried evaluated. The 3 provided in the default code appear to be the best. There seems to be a small range of tau that captures the trajectory; consistent with the notion that replay events happen on smaller timescales. At smaller values of tau, outlier dots outside the trajectory appear. At larger tau, fewer and more inaccurate decoded locations appear.

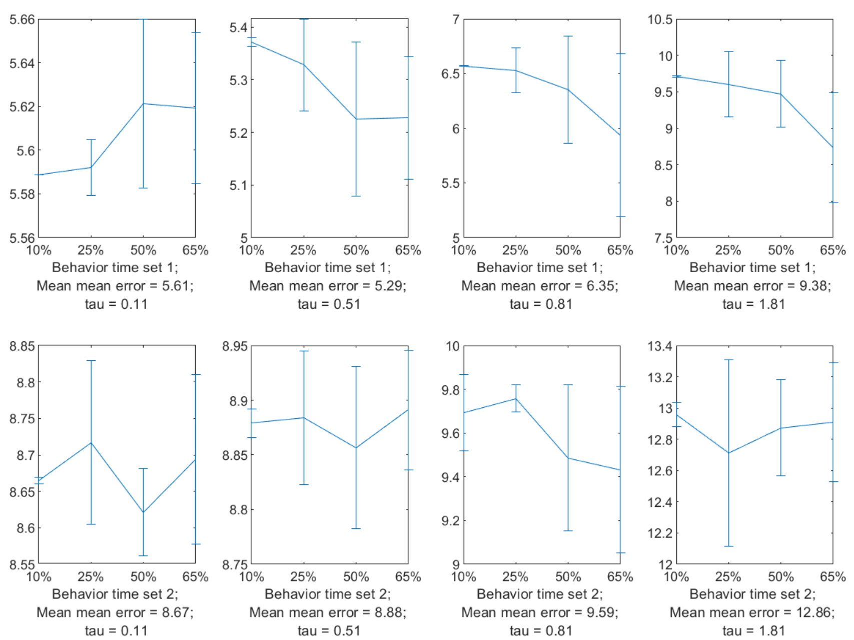


**Part 3: Creative Exploration or Competition** (1 pt)

In this assignment, we have just given you some basic tools to plot tuning curves and visualize the decoded trajectories and locations. With the spike data matrix and the position matrix, you should be able to do many interesting creative exploration. This creative exploration part is required, with 1 point. You should form at least one question of your own and go about analyzing the data to answer it. The question has to be related to this problem set and utilizes the neural data. Thus, it is a creative exploration on neural data analysis.

In this problem, I am exploring a complement to part 2 question 3; rather than dropout, I evaluate the decoding performance with various amounts of noise added to the input. Shown below are the decoding errors across various amounts of noise for 3 different values of tau for the 2 sets of behavior time periods.

The lines represent the average and standard deviation of 10 runs of randomly selecting 10%, 25%, 50%, and 65% of data points to add 1 spike count.




This experiment shows similar trends across the two behavior periods (top row and bottom row), though there is large overlaps in error bars in all cases. Interestingly, there is an decrease in error with more noise added for both behavior cases at tau=0.81.

This decrease in error is slight but worth note considering the large difference in the amount of nonzero values in spike_matrix which is originally quite sparse. It is also worthy to note that this error is higher overall than with smaller values of tau.

Overall, there appears no straightforward relationship with adding noise to input data in terms of tau.

## Below are the mean values shown in the plot printed during execution:

For behavior_time set #1, tau = 0.11, data noised = 0.10, the mean mean error is 5.59
For behavior_time set #1, tau = 0.11, data noised = 0.25, the mean mean error is 5.59
For behavior_time set #1, tau = 0.11, data noised = 0.50, the mean mean error is 5.62
For behavior_time set #1, tau = 0.11, data noised = 0.65, the mean mean error is 5.62
For behavior_time set #1, tau = 0.51, data noised = 0.10, the mean mean error is 5.37
For behavior_time set #1, tau = 0.51, data noised = 0.25, the mean mean error is 5.33
For behavior_time set #1, tau = 0.51, data noised = 0.50, the mean mean error is 5.23
For behavior_time set #1, tau = 0.51, data noised = 0.65, the mean mean error is 5.23
For behavior_time set #1, tau = 0.81, data noised = 0.10, the mean mean error is 6.57
For behavior_time set #1, tau = 0.81, data noised = 0.25, the mean mean error is 6.53
For behavior_time set #1, tau = 0.81, data noised = 0.50, the mean mean error is 6.35
For behavior_time set #1, tau = 0.81, data noised = 0.65, the mean mean error is 5.94
For behavior_time set #1, tau = 1.81, data noised = 0.10, the mean mean error is 9.71
For behavior_time set #1, tau = 1.81, data noised = 0.25, the mean mean error is 9.60
For behavior_time set #1, tau = 1.81, data noised = 0.50, the mean mean error is 9.47
For behavior_time set #1, tau = 1.81, data noised = 0.65, the mean mean error is 8.73

For behavior_time set #2, tau = 0.11, data noised = 0.10, the mean mean error is 8.66
For behavior_time set #2, tau = 0.11, data noised = 0.25, the mean mean error is 8.72
For behavior_time set #2, tau = 0.11, data noised = 0.50, the mean mean error is 8.62
For behavior_time set #2, tau = 0.11, data noised = 0.65, the mean mean error is 8.69
For behavior_time set #2, tau = 0.51, data noised = 0.10, the mean mean error is 8.88
For behavior_time set #2, tau = 0.51, data noised = 0.25, the mean mean error is 8.88
For behavior_time set #2, tau = 0.51, data noised = 0.50, the mean mean error is 8.86
For behavior_time set #2, tau = 0.51, data noised = 0.65, the mean mean error is 8.89
For behavior_time set #2, tau = 0.81, data noised = 0.10, the mean mean error is 9.69
For behavior_time set #2, tau = 0.81, data noised = 0.25, the mean mean error is 9.76
For behavior_time set #2, tau = 0.81, data noised = 0.50, the mean mean error is 9.49
For behavior_time set #2, tau = 0.81, data noised = 0.65, the mean mean error is 9.43
For behavior_time set #2, tau = 1.81, data noised = 0.10, the mean mean error is 12.96
For behavior_time set #2, tau = 1.81, data noised = 0.25, the mean mean error is 12.71
For behavior_time set #2, tau = 1.81, data noised = 0.50, the mean mean error is 12.87
For behavior_time set #2, tau = 1.81, data noised = 0.65, the mean mean error is 12.91