ECE C143A/C243A, Spring 2023 **Homework #3**

Department of Electrical and Computer Engineering Prof. J.C. Kao

University of California, Los Angeles TAs T. Monsoor, R.R. Gore, D. Singla

Due Friday, 5 May 2023, uploaded to Gradescope.
Covers material up to Poisson Processes III.
100 points total.

1. (13 points) You are recording the activity of a neuron, which is spiking according to a Poisson process with rate λ. At some point during your experiment, the recording equipment breaks down and begins dropping spikes randomly with probability p.

   (a) (10 points) Let the random variable M be the number of recorded spikes with the broken equipment. Show that the distribution of M is Poisson($(1-p)\lambda s$). (Hint: If N is a random variable denoting the number of actual spikes, what is $\mathbf{Pr}(M = m|N = n)$?)

   (b) (1 points) What is the rate of the Poisson process in part (a)?

   (c) (2 points) What is the distribution on the number of spikes dropped within a τ second interval?

2. (35 points) Homogeneous Poisson process
   Complete the Jupyter notebook `hw3p2.ipynb`. Print your notebook code and output to a pdf as your submission for this question.

3. (22 points) Inhomogeneous Poisson process
   Complete the Jupyter notebook `hw3p3.ipynb`. Print your notebook code and output to a pdf as your submission for this question.

4. (30 points) Real neural data
   Complete the Jupyter notebook `hw3p4.ipynb`. Print your notebook code and output to a pdf as your submission for this question.

1a) Neuron spiking according to Poisson process with rate λ.

$\rightarrow$ recording equip. breaks, starts dropping spikes randomly with probability p

if N is a RV denoting the actual # of spikes, what is $Pr(M = m \mid N = n)$

RV # of recorded spikes

$$P(N=n) = \frac{(\lambda s)^n e^{-\lambda s}}{n!}$$

$$P(M=m \mid N=n) = \frac{n!}{(n-m)!\, m!} (1-p)^m \, p^{n-m}$$

$$P(M=m \mid N=n) = \frac{P(M=m, N=n)}{P(N=n)}$$

$$P(M=m \mid N=n)\; P(N=n) = \underline{P(M=m, N=n)}$$

$P(M=m)$ = sum across $P(M=m, N=n)$

$$P(M=m) = \sum_{n=0}^{n=\infty} P[M=m, N=n]$$

$$P(M=m) = \sum_{n=0}^{n=\infty} P(M=m \mid N=n)\, P(N=n)$$

$$= \sum_{n=0}^{n=\infty} \frac{n!}{(n-m)!\, m!} (1-p)^m p^{n-m} \cdot \frac{(\lambda s)^n e^{-\lambda s}}{n!}$$

$$= \sum_{n=0}^{n=\infty} \frac{1}{(n-m)!\, m!} (1-p)^m p^{n-m} \left((\lambda s)^n e^{-\lambda s}\right)$$

$$= \sum_{n=0}^{n=\infty} \frac{(\lambda s)^n e^{-\lambda s}}{(n-m)!\, m!} (1-p)^m p^{n-m}$$

I need: $$\frac{[\lambda s (1-p)]^m e^{-\lambda s (1-p)}}{m!}$$

sep.

$$\frac{[\lambda s [1-p]]^m e^{-\lambda s (1-p)}}{m!} \cdot \sum_{n=0}^{n=\infty} \frac{p^{n-m} (\lambda s)^{n-m} e^{-\lambda s p}}{(n-m)!}$$

sum over poisson distribution $= 1$

$$P(M=m) = \frac{\lambda s[1-p]^m e^{-\lambda s(1-p)}}{m!}$$

QED

b) $(1-p)\lambda$

c) $\sim$ Poisson $(p\lambda\tau)$

2c) There should be 800 points on plot

$$8 \left[\circ \quad \right]_{100}$$

list with all the spike time for one trial

go through each trial and

① count the # of spikes

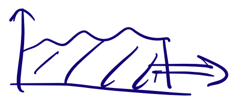
② $\frac{1}{\# spikes} = \lambda$

1 because 1 second

associate the λ to the angle of the test

2d) for each reach angle, plot normalised distribution

count in a class divided by the total # of observations



$= 1$

1. need total # of spikes

spike-counts → each element in spike counts has total spike count for that angle and trial

totalspikes needs to be a $\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_8 \end{bmatrix}$ such that you can normalise based on that value

row, column

8 x 1 →

totalspikes [con] = add spike counts. → spike counts [con, rep]

8 x 100

so I need to go column by column.

$$\text{totalspikes}[K] = \sum_{n=0}^{n=99} \text{spike counts}[k, n]$$

$$\text{totalspikes}[0] = \sum_{n=0}^{99} \text{spike counts}[0, n]$$

first row all columns.

totalspikes [0] = np.sum (spikecounts[0, :])

con   con

∴ totalspikes [con] = np.sum (spikecounts [con, :])

this did not work.

scipy.stats.poisson → "fitting" a poisson distribution to empirical distribution → trying to find the poisson distribution that best describes the empirical distr.

↓

find the PARAMETERS

↙

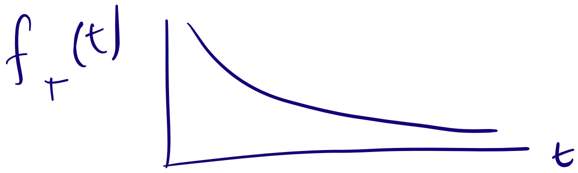
in this case <u>mean</u>

2e) given there are 8 points, one for each angle, I have to take mean and variance of spikes-count [con]

variance →

2f). for each angle, plot <u>normalised dist. of ISIs</u>

I have an $8 \left[\quad \right]_{100}$ with spike times → each has its own trial with spike times

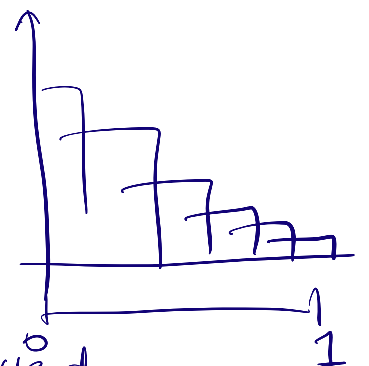


$$f_T(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t \geq 0 \\ 0 & \text{else} \end{cases}$$

np.diff ( )

ISI

This worked with isi_test

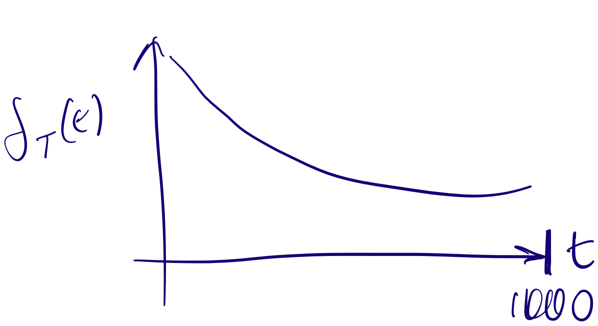


right now they are all plotted separately. I need to take mean of each bin

1 because each trial lasts 1 second ∴ that's the max value this can take

is it in ms?

yes. so should be 1000



not working

new approach

unpack all the 8 [ ] 100 for each row (i.e each angle).

then plot that as a histogram

spikesunpacked will be 8 [ ] 1 here this will be a list of all the spike times for θ_i

then I will do

isis_unpacked$_8$[ ]$_1$

```
for con in range (num_cons):
    isis_unpacked = np.diff (spikesunpacked[con])
```

How to do this:

```
new vectos = [ ]
for con in range (num_con):
    for rep in (num trials):
        new vector[con] += spiketimes[con, rep]
```

↑ does not work

use .exte

2g) spikes unpacked [0 ... 7] list of all the spike times for that reach angle

isis_array [0 ... 7] list of all ISI's for that reach angle.

```
mean_isi = np.zeros(num_cons)
std_isi = np.zeros(num_cons)
coef_v = np.zeros(num_cons)

for con in range(num_cons)
    mean_isi[con] = np.mean(isis_array[con])
    std_isi[con] = np.std(isis_array[con])
    coef_v[con] = st_isi[con] / mean_isi[con]

plt.plot(mean_isi, coef_v)
```