Statistical Methods: Assignment 1

This is the first of three assignments, for which extensive help is available during the tutorials. It is worth 20% of the final course grade.

The deadline for this assignment is the end of Sunday 16 Jan (23:59h). Late work which is submitted up to a day after the deadline will receive a penalty of -20% of the awarded grade, while work submitted 1-2 days after the deadline will receive a penalty of -40% of the awarded grade. Work submitted later than this will not be assessed. The rubric for the assessment of all the assignments, listing the categories assessed and the requirements for each of them, will be provided separately on Canvas.

What you should submit

You should submit your work via Canvas. ***It must be in the form of two Jupyter notebooks, one for part A and one for part B.*** Make sure that you upload the correct files, and check that all the cells run successfully (***and in the correct order***, from start to finish) before you submit them!

When answering each question, use a cell with markdown to briefly explain your approach (a few sentences is fine) along with any assumptions you made. Explanations of your code should be included as comments (or docstrings for functions) in the code cells. Following your code and results obtained should summarise your findings (and interpret them if needed) in a markdown cell following the code cells which produce your results.

Remember that the usual plagiarism rules apply to your work: if you cut-and-paste code from somewhere/someone else you must cite the source (simply replacing variable names is not sufficient to make it your own!). We make an exception to this rule for code from the course's own material, which we allow you to use without citation. We also expect you to help each other, at least early on, and/or be inspired by methods you see online, so programming ***your own version*** (i.e. not cut and pasted) of someone else's method is fine and does not require citation.

Part A (10% of final course grade)

Do the programming challenge at the end of each episode, 1-5. Each challenge contributes 2% of the final course grade.

Part B (10% of final course grade)

For this hypothetical exercise, imagine that you have developed a particle detector to detect a new kind of particle from cosmic accelerator sources, such as the jets produced by formation of black holes in a hypernova. Since the particle is impossible to generate on Earth, calibration of your detector is difficult. In particular, you expect there to be an instrumental deadtime after each particle is detected, during which no other particles can be detected. You expect that this deadtime corresponds to a fixed time interval t_d, but you don't know what the value of t_d is. The situation is illustrated below:

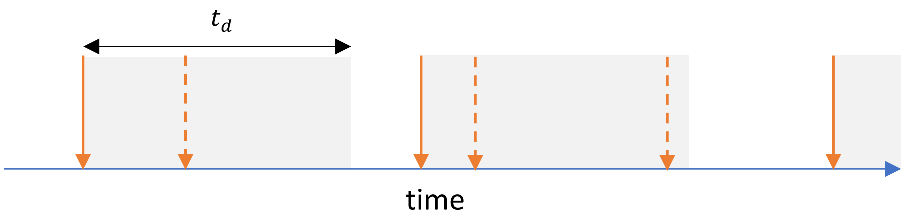

Figure 1 Depiction of effects of deadtime. Particles represented by vertical solid arrows are detected, but trigger a period of deadtime (grey boxes) during which arriving particles (dashed vertical arrows) cannot be detected. Thus in the time shown, 6 particles arrive but only 3 are detected.

The data sets for this problem consist of two text files listing the **detected** particle arrival times (in units of seconds) from your detector for two bursts, both detected over a 1 s interval. The times are measured from a fixed arbitrary start time.

The particles follow a Poisson process. In the absence of deadtime, the 'interarrival' time, i.e. the time difference between a particle arrival time (or an arbitrary time in the past, provided no particle arrived since then) and **the next** particle arrival time in the detector follows an exponential distribution:

$$p(t|\lambda) = e^{-\lambda t} \quad \text{for} \;\; t \geq 0 \quad \text{and 0 otherwise}$$

Where t is the time elapsed and λ is the usual Poisson rate parameter.

Now do the following. All parts are equally weighted:

1. Write down a modified version of the exponential distribution $p(t|\lambda, t_d)$, i.e. which includes the deadtime as a parameter.
2. Use your new probability distribution to measure the **joint** posterior probability distribution of the true rate parameter (i.e. including non-detections due to deadtime) λ and the deadtime t_d for each of the two data sets, plotting both distributions with a 3D plot. For your calculation you can assume a uniform prior for both the rate and deadtime parameters.
3. Obtain the univariate posterior distribution for each parameter and plot them for both data sets. Based on your distributions, comment on whether there is any evidence for a difference in the instrument deadtime between the two data sets.
4. After some careful calibration of your detector with other measurements, you measure a deadtime of 0.097 ± 0.013 s, where the error bar is the 1-σ standard deviation on the measurement and you assume the measurement error is normally distributed. Use this information to recalculate and plot the joint and marginal posterior distributions for both data sets. Comment on how they change from the uniform deadtime prior results, and comment on what they imply for the measured deadtime.

Hints:

- Scipy contains a statistical distribution which you can use for this problem.
- You will find it useful to use broadcasting to calculate a pdf on multiple axes. You can do this using the numpy reshape function to convert 1-D arrays into multidimensional arrays with the other dimensions set to length 1.
- All the relevant code examples and methods are already provided in the web-based episodes for this week.