UNIVERSITY OF CAMBRIDGE

MPhil in Data Intensive Science

Friday 17th November 2023

S1: Principles of Data Science - ***Coursework Assignment***
Matt Kenzie

This is the coursework assignment for S1: Principles of Data Science.

*You should attempt **all** questions. The weighting of marks for each part of the question are given in square,* [. . .]*, brackets in the right-hand margin. One third of the total marks will depend on how well written the report is, how clear and well-explained any figures are, how easy it is to read and run the code, and the spelling, grammar and layout of the report.*

You should write a report of no more than 3000 words to accompany the software you write to solve the problem. Your report should contain two sections, labelled "Section A" and "Section B". Section A should answer parts (a) – (e) with concise answers and short mathematical proofs. It should be no longer than 500 – 1000 words. Section B should answer parts (f) and (g) and contain a more detailed description and justification of your method. You may wish to further partition this into subsections.

There are several different ways to solve this problem. It is more important that you explain and justify your approach, and submit suitable code to perform the task, than it is to get the answer "right".

The software should be submitted to a private `gitlab` *project which can be found at* `https://gitlab.developers.cam.ac.uk/phy/data-intensive-science-mphil/s1_assessment/<YourCRSid>`*. Only you will have access to this project and your access will expire after the deadline for submission, which is **17th December 2023**. The project should contain a* `README.md` *file which describes exactly how to run the code. You should provide a Docker container or a Conda environment so that I can run the code easily. Please include your* `README.md` *file as an Appendix of your report (that will not count towards the word limit) and provide some details of how long it took you to run your code and the specifications of the machine you ran it on. You should ensure your code is well documented with regular comments which explain what your code is doing.*

1 Consider the statistical model below. It depends on a single continuous random variable M, which takes values in the range $M \in [5, 5.6]$. It consists of a smoothly falling background and a peaking signal. The background follows an exponential decay distribution with a probability density

$$b(M;\lambda) = \begin{cases} \lambda e^{-\lambda M} & M \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

where $\lambda > 0$. The signal follows a normal distribution with a probability density

$$s(M;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(M-\mu)^2}{2\sigma^2}\right],$$

where $\sigma > 0$. The fraction of signal is govened by the parameter f such that the total probability density can be written

$$p(M;f,\lambda,\mu,\sigma) = fs(M;\mu,\sigma) + (1-f)b(M;\lambda). \tag{1}$$

(a) Show that the probability density as written in Eq. (1) is properly normalised in the range $M \in [-\infty, +\infty]$. [2]

(b) Given that the random variable M is restriced to the range $M \in [\alpha, \beta]$ derive an expression for the probability density in terms of $\boldsymbol{\theta} = (f, \lambda, \mu, \sigma)$ and the lower and upper limits, (α, β), such that the probability density is properly normalised in the range $M \in [\alpha, \beta]$. [2]

Please note that the f parameter should be interpreted as the fraction of the signal distribution in the range $M \in [\alpha, \beta]$. In other words the signal-to-background ratio in the range $M \in [\alpha, \beta]$ is $f/(1-f)$.

You may find it useful to recall the defintion of the cumulative density function,

$$F(X) = \int_{-\infty}^{X} f(X')dX',$$

which for the normal and exponential decay distributions are,

$$\textit{Normal: } F(X) = \Phi\left(\frac{X-\mu}{\sigma}\right) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{X-\mu}{\sigma\sqrt{2}}\right)\right]$$

$$\textit{Exponential: } F(X) = 1 - e^{-\lambda X}$$

(c) Code these expressions for the probability density in python, in any way you so desire. Plugging in some different values for the parameters $\boldsymbol{\theta}$, use a numerical integration to convince yourself that the total probability density integrates to unity in the range $M \in [5, 5.6]$. [2]

You may use predefined methods in libraries such as `scipy.stats` and `scipy.integrate` to help you.

(d) Assuming that the *true* values of the parameters, $\boldsymbol{\theta} = (f, \lambda, \mu, \sigma)$, take the following values

$$f = 0.1, \quad \lambda = 0.5, \quad \mu = 5.28, \quad \sigma = 0.018,$$

make a plot of the *true* signal, background and total probability distributions all overlaid. [2]

(e) Assuming these true values, generate a single high-statistics sample (containing 100K events) from this probability distribution. Use an estimation method to obtain an estimate for the parameters of the model, along with uncertainties on those estimates, using the generated sample. Make a plot which shows the generated sample, along with the estimates of the signal, background and total probability all overlaid. [2]

(f) Assuming these true values, run a simulation study to estimate how much data you would need to collect in order to "*discover*" the signal at least 90% of the time. In other words generate some pretend datasets from the probability distribution given and then see how easy it is to statistically infer the presence of a signal when you fit those datasets back with the same model. You can assume that the threshold for discovery is 5 standard deviations of a normal distribution, in other words a p-value of less than 2.9×10^{-7}. You should assume that when fitting your simulated samples that you have no *a priori* knowledge of any of the parameters (in other words they should all float). Please justify and explain your approach in your report write up. [10]

Note there are very fast (but poorly documented) evaluations of normal and exponential distributions available in the `numba-stats` *package. You will also find that implementing a binned fit will considerably speed-up your code. As a guide, my solution is not very optimised but ran in about an hour on my fairly high-spec laptop.*

(g) Let us now assume there is a second signal which has the same probability density function as the first signal but with its mean position shifted with respect to the first signal. We can now write our total probability density as

$$p(M) = f_1 s_1(M; \mu_1, \sigma) + f_2 s_2(M; \mu_2, \sigma) + (1 - f_1 - f_2) b(M; \lambda),$$

where f_1 and f_2 are the fraction of each signal, μ_1 and μ_2 their mean positions, σ is the same width for both signals and λ is as before. You can assume that the true values of the parmaeters are,

$$f_1 = 0.1, \quad f_2 = 0.05, \quad \lambda = 0.5, \quad \mu_1 = 5.28, \quad \mu_2 = 5.35, \quad \sigma = 0.018.$$

Determine how much data would be needed in order to determine (at 5 standard deviations) that there are two separate signals (not a single overlapping signal) at least 90% of the time. Please justify and explain your approach in your report write up. [4]

END OF PAPER