DAML Project 1: Parameter Estimation

2023
P.Clarke

Description

This checkpoint is based around a particle decay from which you will measure a parameter related to the matter/anti-matter asymmetry of the Universe.
This relevant decay $X \to D$ has the following PDF:

$$P(t; \tau, \Delta m_s, V) \propto (1 + V \sin(\Delta m t)) \times \exp(-t/\tau)$$

where

- t is the observable quantity - the decay time of each decay;
- τ is a lifetime parameter;
- Δm is a mass difference parameter which leads to sinusoidal oscillations superimposed on the exponential decay.
- V is a parameter which measures matter/anti-matter asymmetry and has the value zero if the universe is symmetric (which we know it isnt !);

The nominal values of the parameters are

- $\tau = 1.5$ (arbitrary units)
- $\Delta m = 20.0$ (arbitrary units)
- V = 0.1

Part 1: Estimating statistical precision [5 marks]

Use the method of multiple pseudo-experiments (toy Monte Carlo event generation) to determine the expected statistical precision with which one could measure each of the parameters with 10000 events. You should:

- Use a time range of [0,10] to generate your events
- Generate and make a histogram of a single dataset first, in order to check your PDF is working. You may wish to increase V temporarily to make the oscillations term more visible.
- Use a maximum likelihood fit to determine the precision expected for each of the parameters $\tau, \Delta m$ and V (*hint: you need enough toy datasets to perform the required statistical analysis, typically at least 100*).
- Determine the bias (if any) for each parameter. To do this you need to know how to estimate the bias as well as the precision of the bias, in order to know if it is significant.

Hint: Because this involves a lot of generation of events (100 toys @ 10000 events) it is better to use a much smaller number to develop your code. Then it will run much faster. It is suggested you start with 10 toys @ 1000 events, to make sure code runs, then 50 toys @3000 events for main development, and then at least 100 toys @ 10000 events when the code is mostly all working and to produce your final results.

Part 2: Estimating possible bias due to an unaccounted for background [5 marks]

In this part you will use the same signal PDF as in PART1. We hypothesise that the data has an additional background component within it that we do not know about. In this example the background will be another exponential with a much longer lifetime: it is 8 units (compared to 1.5 for the signal). For this exercise you will:

- Generate 5 data sets (each set being 10000 events), each with a different exponential background fraction (with a lifetime of 8) added to the signal. The fractions will be [0.0, 0.1, 0.2, 0.3, 0.4]. Use a time range of [0,10] to generate your events;
- Fit to each of these data sets as if you do not know about the background. In other words you will use only the signal PDF used in PART1 for the fit;
- Create plots showing how the fitted value of each parameter τ, Δm and V varies as a function of the background fraction;
- Comment on any observed bias/trend and whether it is significant. To do this you need to compare any change seen in a parameter to its error.

Remember, you are only asked to generate a single toy data set for each background fraction in order to limit the time needed to run your code. In reality one would generate more toys at each point or run with more events.

Hint To generate the data in this part you need a composite PDF containing both the signal PDF and the background PDF. The easiest way to do this is create a new PDF as the sum of
*(1-F)*SignalPDF + F*BackgoundPDF*
where F is the fraction of background. If each of your PDFs is correctly normalised, then so wil the compound PDF be automatically.

General Instructions for Reports

It is strongly recommended that the project reports are submitted in the jupyter notebook format. It is expected that you are comfortable with writing, running and annotating code within a jupyter notebook, as this is the norm by now.

Annotation and Commentary: It is important that all code is annotated and that you provide brief commentary at each step to explain your approach. We expect well-documented submissions (jupyter notebooks or other format), not an unordered collection of code snippets. You can also include any failed approaches if you provide reasonable explanation.

Unlike weekly checkpoints where you were being guided towards the "correct" answer, the projects are by design more open ended. It is, therefore, necessary to give some justification for choosing one method over another.

Project reports are not in the form of a written report so do not provide pages of background material. Only provide a brief explanation for each step. Aim to clearly present your work so that the markers can easily follow your reasoning and can reproduce each of your steps through your analysis.

If you are using jupyter notebooks: To add commentary above (or below) a code snippet create a new cell and add your text in markdown format. Do not add commentary as a code comment in the same cell as the code.

A fraction of the mark for each exercise is allocated to coding style and clarity of comments and approach (how much exactly is in the discretion of each lecturer).

It is important your code is fully functional before it is submitted or this will affect your final mark. If you are using jupyter notebooks, when you are ready to submit your report perform the following steps:

- In Jupyter run Kernel >Restart & Run All to ensure that all your analysis is reproducible and all output can be regenerated
- Save the notebook, and close Jupyter
- Tar and zip your project folder if you have multiple files in a working directory. (If you are not familiar with these operations make a backup of your files before executing any commands.) You are free to include any supporting code. Make sure this belongs in the project folder and is referenced correctly in your notebook. Do not include any of the input data.
- Submit this file or zipped folder through Learn/Turnitin. In case of problems or if your compressed project folder exceeds 20 MB (first make sure you are not including any input files, then) email your submission to the course administrator (Kieran Brodie).