TIF345/FYM345: Project 3

A Galton board on a rocking ship

30 Points. Due date/time: 2023-12-21 23:59

Questions and the requests for clarifications should be addressed to Arkady Gonoskov (arkady.gonoskov@physics.gu.se)

In this project your task is to study the properties of a Galton board that is virtually provided via Python module. The board consists of 31 rows of pegs. We are given with information that the beads possess a peculiar property of inertia that is described by a hypothesis: if a bead falls to the right/left of a peg then it starts spinning clockwise/counterclockwise and this makes it more likely that the bead falls to the same side at the next peg. We can model this by a coefficient α that quantifies the bias due to the previous move. We assume that earlier moves (more than one peg earlier) have no effect and that the bias is zero at the first peg (see the model below). We want to determine the most probable value of α by a series of experiments, in each of which we get the distribution of beads in the bottom of the board. The difficulty is that we happen to carry out the experiments on a slowly rocking ship so that in each experiment there is an additional bias s that increases the chance that, in each hit, every bead falls in one particular direction. The bias due to the current slope of the ship is constant during a single experiment – i.e. s is the same for all events in a single experiment. The value of s quantifies the slope of the ship and varies uncontrollably from experiment to experiment. According to our model, the probability of a bead falling to the right of a peg P_+ and the probability of falling to the left of a beg P_- are described by the equation:

$$P_{\pm} = 0.5 \pm (\alpha M + s),$$

where $M = 0$ at the entrance, $M = -0.5$ if the bead bounced off the previous peg to the left (so it arrives at the present peg from the right) and $M = 0.5$ if the bead bounced off the previous peg to the right (so it arrives at the present peg from the left).

You are given with a python module "board" (upload `board.so` to the working directory and use Python 3.10) that contains a function `experiment()` for making virtual experiments. The function returns a `numpy` vector that contains the number of beads in each cell in the bottom of the board. The slope s is randomly assigned in each experiment. You can assume that $s \in [-0.25,\ 0.25]$ and $\alpha \in [0, 0.5]$. You can call the function, i.e. carry out experiments, as many times as you need.

Task: Determine the most likely value of α and estimate the credible interval (CI). Explain what hampers further decrease of CI in your solution. Describe possible improvements and implement the most useful of them.

Methods: You are supposed to develop ABC routine and then improve it by using an artificial neural network (NN) to eliminate latent variable s. Consider the following plan:

1. Write your own simulator for the described model, i.e. the function that generates the outcome of an experiment for any given values of α and s. Generate a set of simulated results for various values of parameters.

2. Consider the outcomes of the simulator. Propose a summary statistic and determine the reasonable size of the kernel for the Approximate Bayesian Computation (ABC). Develop the standard ABC routine using uniform random proposal generation for s.

3. Train an NN to solve the inverse problem: determine the value of α and s based on the simulated result. Quantitatively characterize the errors made by the NN with respect to α and s.

4. Modify the ABC routine using the trained NN for the generation of proposals for s.

5. Consider arranging a chain of transformations: the posterior sampled by the ABC procedure is used as prior for the next ABC procedure and so on. As always, it can be useful to model the prior/posterior by some function (e.g. normal distribution) with parameters being re-determined by the generated samples after each ABC procedure. Each ABC procedure can be based on one or many new experimental results (to avoid data reusing). Discuss what factors prevent further decrease of CI.
6. There are many improvements that one can implement for this project. Try to implement the basic plan first and then consider what can be improved.

Literature:

1. The Galton board: https://en.wikipedia.org/wiki/Bean_machine

2. ABC: Sisson, Fan and Beaumont, *Handbook of Approximate Bayesian Computation* (2019)

3. The course textbook, Chapter 10.

Comments:

1. Please verify that you can generate data with the module as soon as possible, but no later than by lab session 2023-12-13 so that technical difficulties can be resolved well before you approach the due date. You might need to use Python environment to set version to 3.10 (see https://realpython.com/intro-to-pyenv/#virtual-environments-and-pyenv).

2. The project is designed so that the runtime for the basic result should be no more than several minutes. If you want to try to achieve a better accuracy, please consider using your PC instead of the provided Jupyter hub.

3. Please try to tackle the project as far as you can by the lab session on 2023-12-13. During the session you will have the opportunity to ask questions in case you face difficulties.