Report on Project 3: A Galton board on a rocking ship

Erik Karlsson Öhman, Hampus Hansen

August 5, 2025

1 Introduction

The Galton board modeled in this task consists of 31 rows of pegs, and $1\,000$ beads. The beads are dropped from the same location above the first row, and once they pass the final row they are collected in 32 slots. The distribution of beads in the slots is normally distributed. The variance of the normal distribution is affected by the assumed property of inertia that the beads possess. This property is described by the hypothesis:

a bead falling to the right/left of a peg starts to spin clockwise/counterclockwise, increasing the likelihood of the bead falling on the same side of the next peg,

and this bias is modeled by parameter α. The bias is assumed to be zero at the start, and only affected by the previous peg [1].

The purpose of this report is to infer the most likely value of parameter α, using data from both experiments and simulations. However due to the rocking of the ship, each experiment is subjected to a random slope $s \in [-0.25, 0.25]$, effecting both variance and the mean of the distribution. This latent parameter complicates the inference of α. Three different methods are implemented to adress this: standard ABC, ABC with ML-eliminated latent variable and chain of ABC routines.

2 Methods

Approximate Bayesian computation (ABC) is a method used if the calculation of the data likelihood is intractable, which makes it unfeasible to compute the posterior through Bayes formula

$$p(\theta|y_{\text{obs}}) = \frac{p(y_{\text{obs}}|\theta)\pi(\theta)}{\int_\theta p(y_{\text{obs}}|\theta')\pi(\theta')d\theta'}. \tag{1}$$

To avoid explicit computation of the likelihood, ABC is implemented by generating a proposal, $\theta^* \propto \pi(\theta)$, where π is the prior. A simulation of the system allows generation of outcomes for this proposal, y^*. If $y^* = y_{\text{obs}}$ the proposal is accepted and then a new proposal is drawn from the prior. This process is repeated until an approximate posterior distribution has been generated.

The kernel relaxes the need for perfect matches, which is impossible in reality. The kernel weights closer matches heavier, and punishes worse matches. A Gaussian kernel is used for this project

$$K = \exp\left(-\left(\frac{S(y_{\text{obs}}) - S(y^*)}{h}\right)^2\right), \tag{2}$$

where S is the summary statistic function. For this problem, the chosen summary statistic is

$$S(y) = \mu + \sigma^2, \tag{3}$$

where μ is the mean, and σ^2 is the variance of the distribution of beads in slots. The parameter α will only affect the variance whilst the slope s will affect both the mean and the variance. The size of the kernel is h, which directly effects the acceptance ratio. Too small acceptance ratio wastes computational resources, whilst too large acceptance ratio broadens the approximate posterior distribution. This size of the kernel is set to $h = 0.7$ for all subsequent tasks, such that the acceptance ratio is around $5 - 10\,\%$, which is considered a fair trade-off.

The generation of proposals requires simulated data, given parameters $\theta = \alpha, s$. The probability of a bead falling to the left (P_{-}) or to the right P_{+} is given by

$$P_{\pm} = 0.5 \pm (\alpha M + s), \tag{4}$$

where $M = 0.5$ if the bead has arrived from the right, $M = -0.5$ if the bead has arrived from the left and $M = 0$ at the start [1].

2.1 Standard ABC routine

For the standard ABC routine the priors of both parameters are modeled by uniform distributions, $s^* \sim \mathcal{U}(-0.25, 0.25)$ and $\alpha^* \sim \mathcal{U}(0, 0.5)$. This is motivated by the fact that we have no real prior knowledge about the problem at hand.

2.2 ABC with latent variable elimination with ML

Given this system, the slope of the Galton board is a latent parameter. To improve the inference of α without knowing s we will employ an ABC with ML-based latent variable elimination. It is based on determining s from the outcomes independently of α through machine learning. The data set consists of $100\,000$ simulations, each with a corresponding $\{\alpha, s\}$-configuration randomly drawn from uniform priors $\mathcal{U}(0, 0.5)$ and $\mathcal{U}(-0.25, 0.25)$ respectively. The data set is split into $25\,\%$ validation data and $75\,\%$ training data. The simulated outcomes y_i from the training data are then fitted to the parameters α and s, using a MLP-regressor from the `sklearn.neural_network`-library. The network consists of one input layer of 32 neurons, 5 fully connected hidden layers of 64, 32, 16, 8 and 4 neurons respectively and an output layer of 2 neurons, which is visualized in Figure 1. $\tanh$ was used as activation function for the network and stochastic gradient descent was used to train the network.

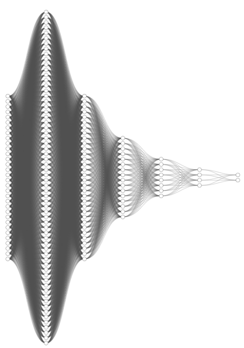


Figure 1: Architechture of the neural network.

The error of the neural networks prediction of s, $s_{\text{error}} \equiv s_{\text{predicted}} - s_{\text{data}}$ is approximately normally distributed, as seen in Figure 2. The standard deviation of the normal distribution is given by the root mean square error of the neural networks prediction of s, $\sigma_s^{\text{RMSE}} = 0.011$. Now instead of drawing s^* from a uniform distribution, we instead let $s^* \sim \mathcal{N}(s^{\text{NN}}, \sigma_s^{\text{RMSE}})$, where s^{NN} is the neural networks prediction of s for an observed board state y_{obs}.

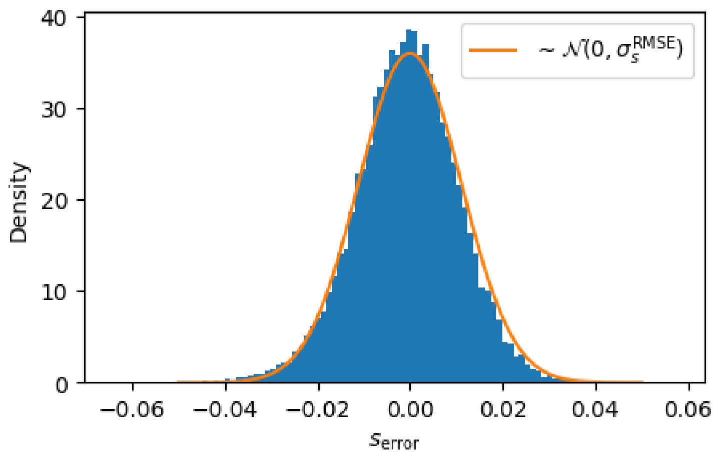


Figure 2: Distribution of $s_{\text{error}} \equiv s_{\text{predicted}} - s_{\text{data}}$.

2.3 Chain of ABC routines

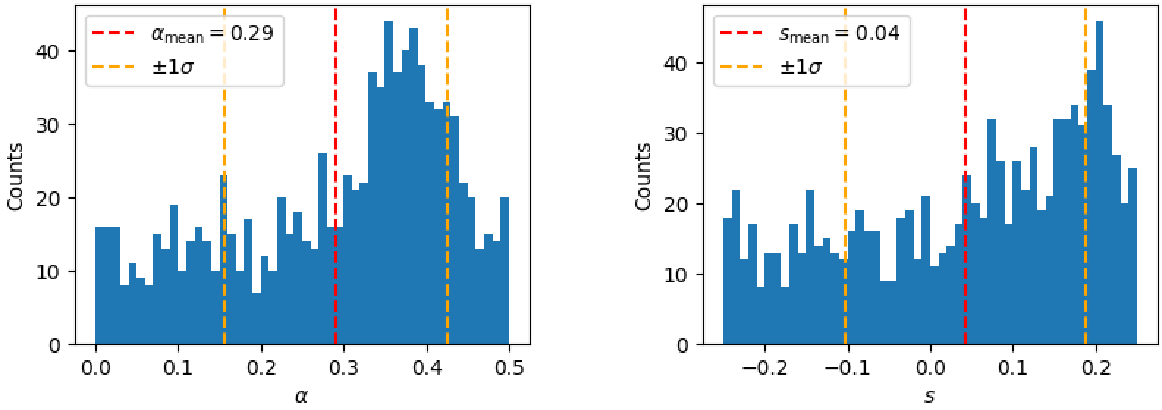
The inference is further improved by multiple chained ABC routines, where the subsequent prior is set as the approximate posterior of the former ABC routine. Specifically, the subsequent prior is thus a normal distribution characterized by the mean and variance extracted from the approximate posterior of the former ABC. This procedure is repeated 10 times, where each approximate posterior distribution consists of $1\,000$ accepted proposals. The first prior is initialized to a uniform distribution. The experimental set of data y_{obs} is randomly drawn from the $10\,000$ available sets to avoid data reusing.

3 Results and discussion

The result from the three different methods described above is presented and discussed in the following section.

3.1 Standard ABC

The resulting approximate posterior distributions of α and s for the standard ABC are presented in Figure 3. The posteriors are constructed from $1\,000$ accepted proposals with acceptance ratio $4.2\,\%$. The approximate posteriors yields $\alpha_{\text{mean}} = 0.29$ and $\sigma_\alpha = 0.14$. The inference of α suffers since the latent parameter s also need to be inferred in standard ABC. We know that the slope is randomly assigned in each experiment, which is not properly reflected in Figure 3b. The large uncertainty of s propagates into the inference of α, resulting in a worse and less certain prediction of the parameter of interest.



(a) Approximate posterior distribution of α.

(b) Approximate posterior distribution of s.

Figure 3: Standard ABC generated approximate posteriors for α and s, where the red and orange dashed lines mark the mean and $\pm 1\sigma$-level respectively.

3.2 ABC with latent variable elimination with ML

The NN predicts values of α and s given simulated data. The root mean square errors of the NN-model is $\text{RMSE}_\alpha = 0.038$ and $\text{RMSE}_s = 0.011$. The predictive capabilities of the NN is presented in Figure 4, where a clear positive correlation between the data and the prediction is seen. That the network is predicts s somewhat better than α may be due to the fact that both parameters affect the shape of the resulting distribution, whilst s alone controls the offset of the distribution, i.e. the mean.

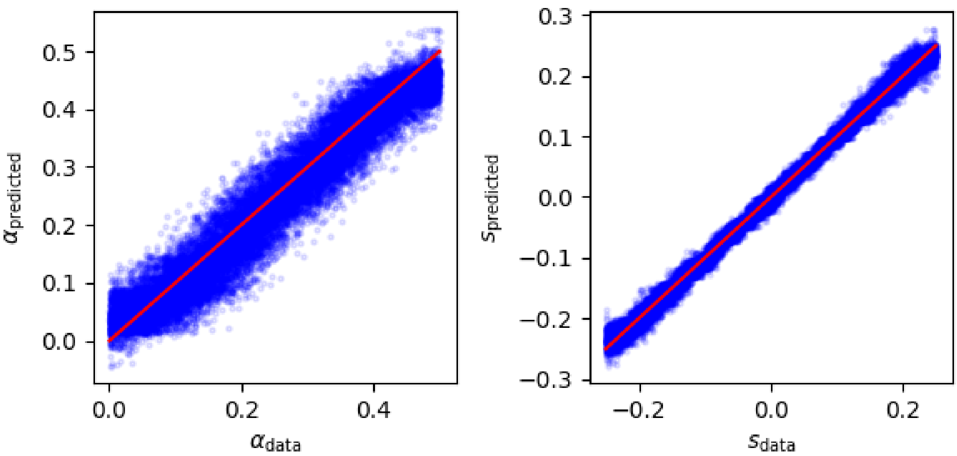


Figure 4: NN-predictions for s and α plotted against the validation set.

Using the NN to eliminate the latent parameter s results in drastically improved inference of α, as seen in Figure 5, with $\alpha_{\text{mean}} = 0.35$ and $\sigma = 0.081$.

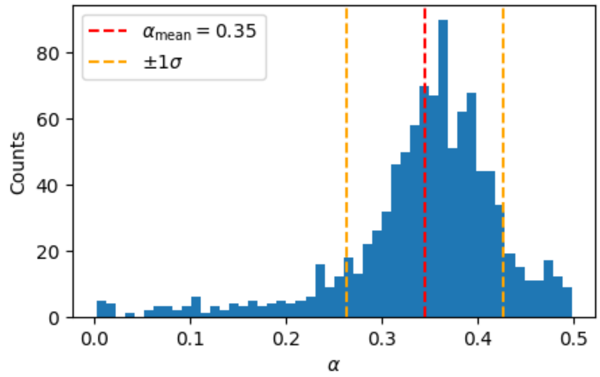


Figure 5: Approximate posterior distribution of α generated by the ABC routine with ML-based latent variable elimination.

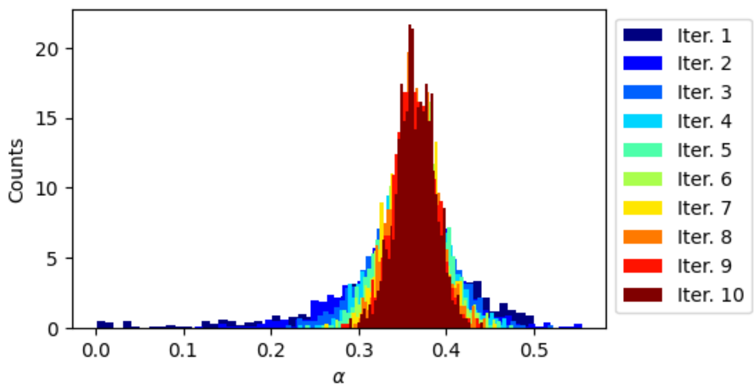
3.3 Chain of ABC routines

The approximate posterior of α generated by a chain of 10 ABC routines can be seen in Figure 6. The final (iteration 10) approximate posterior is seen in maroon in Figure 6a, centered around $\alpha_{\text{mean}} = 0.36$ with $\sigma_\alpha = 0.022$. This result shows a drastic improvement compared to the standard ABC and the ABC with latent variable elimination using ML.

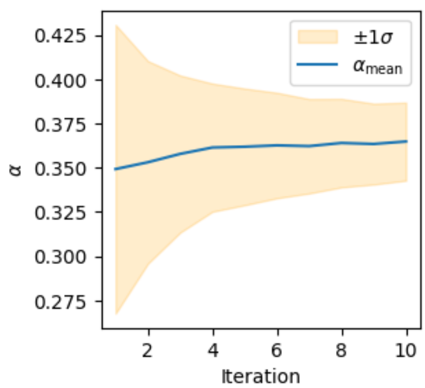
From Figure 6b we see that the credible interval decreases significantly for the first few iterations after which the rate of decrease slows down. Further decrease of the credible interval is likely inhibited by several factors.

One of the factors is the error from the NN-predictions. Another major factor is the intrinsic information loss associated with the reduction of dimensionality from the summary statistic. All information can not be conserved while reducing the dimensions to a few statistics. More complex summary statistics could be implemented, which could decrease the credible interval further.

The kernel size is also a likely factor to the prevention of a further decreased confidence interval, since h affect the acceptance ratio and thus the variance of the sampled distributions. A potential mitigation of this could be to dynamically decrease the size of the kernel as the ABC-chain progresses. This would result in smaller acceptance ratios, and thus more narrow posterior distributions, leading to a further decreased confidence interval.



(a) Approximate posteriors for each step of the 10-chain ABC-method.



(b) Converging α_{mean} with $\pm 1\sigma$-interval.

Figure 6: Converging tendency of α for the chained ABC for 10 iterations, with the approximate posteriors (a) and mean and standard deviation of α (b).

Further improvements could be made to the procedures above. The property of inertia that the beads possess is described by the hypothesis described in the introduction. This hypothesis is assumed to be true, and the simulations are modeled thereafter. Hypothesis testing could be performed to check how valid this hypothesis is. One could also model simulations where the bias is effected by moves more than one peg earlier. The outcome of these changes will depend on the nature of the experiments.

References

[1] A. Gonoskov, *TIF345/FYM345: Project 3 - A Galton board on a rocking ship*. Gothenburg, Sweden: Chalmers University of Technology, Nov. 26, 2024.