S2: Statistics for Data Science

Max Talberg

March 9, 2024

Contents

0.1 The Lighthouse Problem

0.1.1 (i) Trigonometric Analysis of the Lighthouse Problem

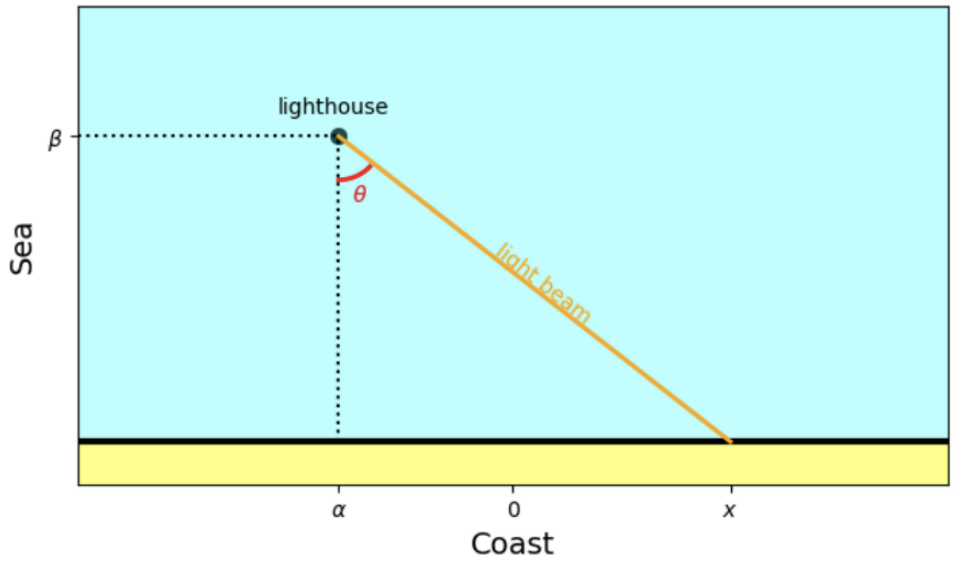

Figure 1: Diagram of the setup of the lighthouse problem.

Using the geometry of the problem illustrated in Figure 1, the trigonometric relationship between the lighthouse's position at (α, β), the angle of the light beam θ, and the point on the coastline x can be established.

From trigonometric principles the tangent of angle θ is the ratio of the opposite side (horizontal distance to x) to the adjacent side (lighthouse's height), yielding:

$$\tan(\theta) = \frac{\text{opposite}}{\text{adjacent}} = \frac{x - \alpha}{\beta} \tag{1}$$

$$\beta \tan(\theta) = x - \alpha \tag{2}$$

$$x = \beta \tan(\theta) + \alpha \tag{3}$$

0.1.2 (ii) Derivation of the Likelihood Function for Flash Location

The angle of a flash is denoted by θ and is uniformly distributed in the range $-\pi/2 < \theta < \pi/2$.

Uniform Distribution of θ: The probability density function (PDF) for θ is given by,

$$P(\theta) = \mathbb{1}_{(-\pi/2,\pi/2)}(\theta)\frac{1}{\pi}, \tag{4}$$

where $\mathbb{1}_{(-\pi/2,\pi/2)}(\theta)$ is an indicator function ensuring that $P(\theta)$ is defined only within the specified range. This reflects the assumption that flashes are equally likely to occur in any direction within this range.

Transformation to x: The likelihood of observing a flash at location x, given α and β, involves transforming the PDF from θ to x. This is based on the transformation law,

$$P(x|\alpha, \beta)dx = P(\theta|\alpha, \beta)d\theta, \tag{5}$$

$$P(x|\alpha, \beta)dx = P(\theta|\alpha, \beta)\frac{d\theta}{dx}dx. \tag{6}$$

Equation 3 relates θ and x, which rearranges to give $\theta = \arctan\left(\frac{x-\alpha}{\beta}\right)$. Differentiating this with respect to x gives,

$$\frac{d\theta}{dx} = \frac{\beta}{\beta^2 + (x-\alpha)^2}. \tag{7}$$

Substituting this into the transformation law yields the PDF of x,

$$P(x|\alpha,\beta)dx = P(\theta|\alpha,\beta)\frac{\beta}{\beta^2 + (x-\alpha)^2}dx, \tag{8}$$

$$P(x|\alpha,\beta) = \frac{1}{\pi}\frac{\beta}{\beta^2 + (x-\alpha)^2}. \tag{9}$$

Conclusion: The derived PDF $P(x|\alpha,\beta)$ represents the likelihood $\mathcal{L}_x(x|\alpha,\beta)$ of observing a flash at location x, given the parameters α and β. This likelihood is given by,

$$\mathcal{L}_x(x|\alpha,\beta) = \frac{1}{\pi}\frac{\beta}{\beta^2 + (x-\alpha)^2}. \tag{10}$$

This represents the PDF of the Cauchy distribution with location parameter α and scale parameter β. The Cauchy distribution is a pathological function as both its mean and variance are undefined.

0.1.3 (iii) Most Likely Flash Location

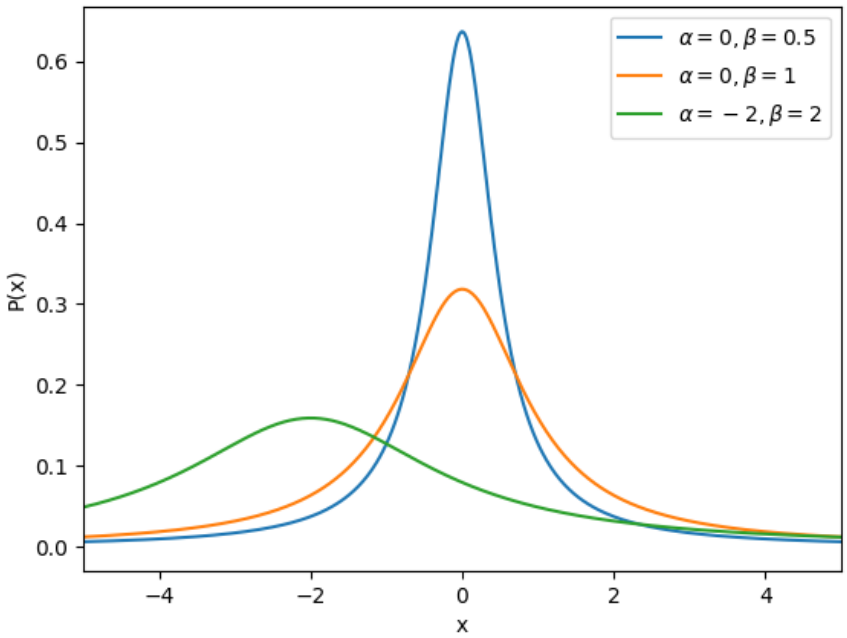

Figure 2: The graph displays three Cauchy distributions with varying location (α) and scale (β) parameters. The location parameter α shifts the peak of the distribution along the x-axis, representing the most frequent flash coordinates of the lighthouse. The scale parameter β influences the spread of the distribution, with higher values indicating a larger spread and suggesting a greater distance from the shore to the lighthouse. The blue line ($\alpha = 0$, $\beta = 0.5$) shows a narrow spread centered at zero, the orange line ($\alpha = 0$, $\beta = 1$) represents the standard Cauchy distribution with a wider spread, and the green line (α = -2, $\beta = 2$) shows the widest spread, shifted to the left.

Cauchy distribution: The colleague is correct that the most likely location for any flash to be received is α. This is due to the symmetric properties of the Cauchy distribution, which can be seen in Figure 2, around its location parameter α. At this median and modal point the PDF reaches its maximum making α the most likely value for any single observation. Using the sample mean, $(1/N)\sum_k x_k$, isn't a good estimator for α because the heavy tails of the Cauchy distribution leads to an undefined mean, as the integral meant to calculate the mean fails to converge. An improved estimator for α would be the Maximum Likelihood Estimate (MLE). The MLE of α is derived by setting up and maximising the likelihood function based on the Cauchy distribution's PDF. This process leads to the sample median being the MLE for α, due to the symmetric nature of the Cauchy distribution. The median is more robust in the presence of outliers and extreme values, which are characteristic of the Cauchy distribution.

Analytical comparison: Investigating the expectation value of a simplified expression of the likelihood function, equation 10, proves the undefined nature of the sample mean,

$$\mathbb{E}_x[x] = \int_0^L x \frac{1}{(a - x^2)} dx, \tag{11}$$

$$\mathbb{E}_x[x] = \left[ln(x - a) - \tfrac{a}{x-a}\right]_0^L. \tag{12}$$

It is clear the sample mean tends towards lnL as L becomes large. This property from the continuous distribution carries over to the behavior of the sample mean when computed from discrete samples drawn from a Cauchy distribution. It is clear this function does not possess a well defined mean and does not converge as more data is collected. Therefore, the sample mean is not a good estimator of the most likely location for a flash to be received.

Due to the symmetric property of the Cauchy function the most likely location for a flash to be received is better found using the median or mode represented by the MLE. The total likelihood of all observations is,

$$\mathcal{L}_x(\{x_k\}|\alpha, \beta) = \prod_k^n \frac{1}{\pi} \frac{\beta}{\beta^2 + (x - \alpha)^2}. \tag{13}$$

Taking the natural logarithm of the total likelihood gives the log-likelihood function,

$$\log \mathcal{L}_x(\{x_k\}|\alpha, \beta) = \sum_k^n \log\left(\frac{1}{\pi} \frac{\beta}{\beta^2 + (x - \alpha)^2}\right). \tag{14}$$

This simplifies to,

$$\log \mathcal{L}_x(\{x_k\}|\alpha, \beta) = \sum_k^n \left(\log(\beta) - \log(\pi) - \log[\beta^2 + (x_k - \alpha)^2]\right), \tag{15}$$

$$\log \mathcal{L}_x(\{x_k\}|\alpha, \beta) = n\log(\beta) - n\log(\pi) - \sum_k^n \log[\beta^2 + (x_k - \alpha)^2]. \tag{16}$$

To find the MLE for α the derivative of the log-likelihood function is taken with respect to α and set equal to zero,

$$\frac{\partial}{\partial \alpha} \log \mathcal{L}_x(\{x_k\}|\alpha, \beta) = 2\sum_k \frac{x_k - \alpha}{\beta^2 + (x_k - \alpha)^2} = 0. \tag{17}$$

It is clear to see for a single observation x the MLE for α is itself x. The case for multiple observations is significantly more complex is not a single observed value x_k but a value that accommodates the distribution of values, which for the Cauchy distribution tends to the median.

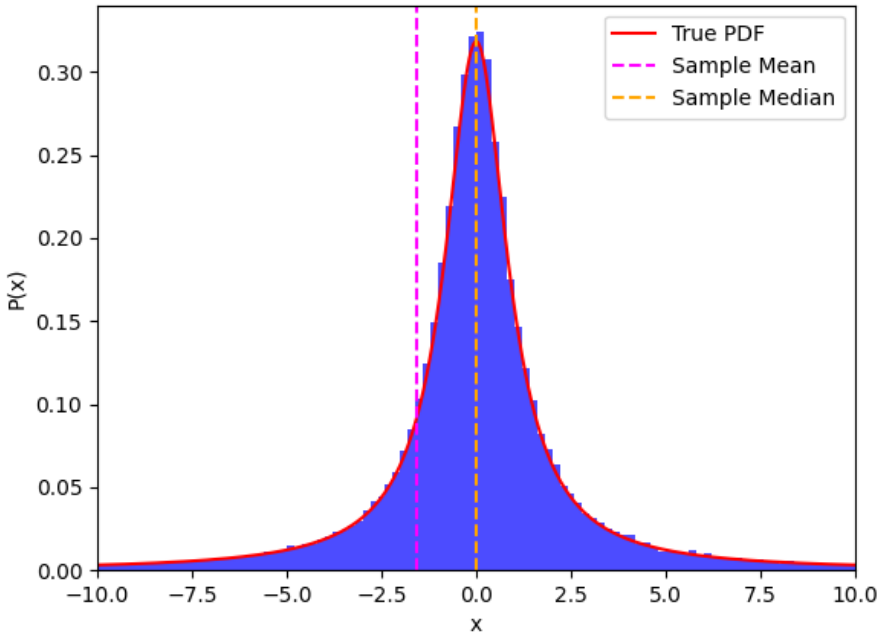

Figure 3: The histogram represents the simulated distribution of 100,000 random data points (flashes) generated from a Cauchy distribution with location parameter $\alpha = 0$ and scale parameter $\beta = 1$. Overlaid on the histogram is the true PDF of the Cauchy distribution, depicted by the red curve. The vertical dashed lines indicate the calculated sample mean (magenta) and sample median (orange) of the data set. These lines illustrate the concept that for distributions with heavy tails like the Cauchy distribution, the median can be a more robust measure of the most likely location than the mean.

Empirical comparison: Evidenced by Figure 3 the median's closer alignment with the peak of the true PDF compared to that of the mean, supports the analytical and theoretical evidence that the MLE is a better estimator for α than the sample mean.

0.1.4 (iv) Selecting a Suitable Prior for α and β

The Bayesian prior represents the state of knowledge before any data. There is no information about the location of the lighthouse, hence a non-informative uniform distribution over the rectangular region spanning horizontally from a to b and vertically from c to d satisfies this ignorance,

$$\pi(\alpha, \beta) = \begin{cases} ((b-a) \times (d-c))^{-1} & \text{for } a < \alpha < b \text{ and } c < \beta < d. \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

0.1.5 (v) Stochastic Samples from Posterior Distribution

Stochastic sampling algorithm: The Markov chain Monte Carlo (MCMC) algorithm used here is the Hamiltonian Monte Carlo (HMC) sometimes known as the hybrid Monte Carlo [1]. The HMC utilises Hamiltonian dynamics to explore high-dimensional spaces and is utilised here to draw samples from a target posterior distribution[2].

The HMC begins by augmenting the state space with pairs of position and momentum variables. A canonical distribution is composed of the target distribution (representing the position) and an easy-to-sample distribution (representing momentum), typically a multivariate Gaussian. These terms correspond to the potential and kinetic energy, together forming the Hamiltonian. The trajectory of the system is explored using leapfrog integration, which alternates between updating the position and momentum. This integration proposes a new state which is subject to a Metropolis-Hastings acceptance step. If the proposed step is rejected, the algorithm

retains the current state. Momentum terms are discarded and resampled from their distribution. This process of drawing momentum, simulating Hamiltonian dynamics and performing the Metropolis-Hastings acceptance step is iteratively repeated.

HMC is advantageous because it does not require a proposal distribution.Similarly the small time steps in the leapfrog integration conserves the Hamiltonian well, resulting in a high acceptance rate. The HMC uses gradient information and Hamiltonian dynamics to explore parameter space well, compared to undesirable random walk behaviour from other Sampling algorithms.

The HMC does however have drawbacks, in that it can only be applied to smooth target distributions because the algorithm requires derivatives of the target distribution. Despite the heavy tails and the undefined mean the Cauchy function is a smooth distribution, which means derivatives are well defined.

During implementation of the algorithm there are many input parameters to tune, which is a drawback for HMC. A variation of the HMC is the No U-Turn Sampler (NUTS). NUTS provides an automated way to determine the step size L in the leapfrog integration. This results in one less parameter to tune when implementing HMC[3].

Software implementation: The HMC algorithm was implemented using TensorFlow [4]. TensorFlow performs automatic differentiation of the log-posterior,then leapfrog integration followed by the Metropolis acceptance step. TensorFlow is a robust package for automatic differentiation and numerical methods, the package also provides utilities to aid the tuning and adaptation of the sampling process.

In Python, this involves initialising a NUTS kernel consisting of the target log probability and the step size, the choice of this kernel omits the need for the number of leapfrog steps. Additionally an adaptive step size wrapper was used, which for a percentage of the burn-in steps explores the parameter space and adaptively selects an appropriate step size. The MCMC sample chain is then initialised with the NUTS kernel and adaptive step size wrapper, the initial state and the desired number of samples. The sample chain is then run in parallel with 8 other chains at random starting points; note that there are additional parameters in this implementation that have been omitted for clarity.

Posterior distribution: The posterior distribution in the range $a < \alpha < b$ and $c < \beta < d$ is given by,

$$P(\alpha, \beta|\{x_k\}) = \frac{\mathcal{L}_x(\{x_k\}|\alpha, \beta)\pi(\alpha, \beta)}{Z}, \tag{19}$$

$$P(\alpha, \beta|\{x_k\}) = \frac{1}{Z}\prod_k^n \frac{1}{\pi}\frac{\beta}{\beta^2 + (x-\alpha)^2}\frac{1}{(b-a)(d-c)}. \tag{20}$$

The HMC method works with the unnormalised log-posterior to ensure numerical stability, the evidence term can be omitted as this disappears in the Metropolis Hastings acceptance (ratio) step,

$$P(\alpha, \beta|\{x_k\}) \propto \log\left(\prod_k^n \frac{1}{\pi}\frac{\beta}{\beta^2 + (x_k-\alpha)^2}\frac{1}{b-a)(d-c)}\right), \tag{21}$$

$$P(\alpha, \beta|\{x_k\}) \propto n\log(\beta) - n\log(\pi) - \log((b-a)(d-c))) - \sum_k^n \log[\beta^2 + (x_k-\alpha)^2]. \tag{22}$$

This posterior distribution was modelled in Python using a HMC provided by TensorFlow, the results are below.

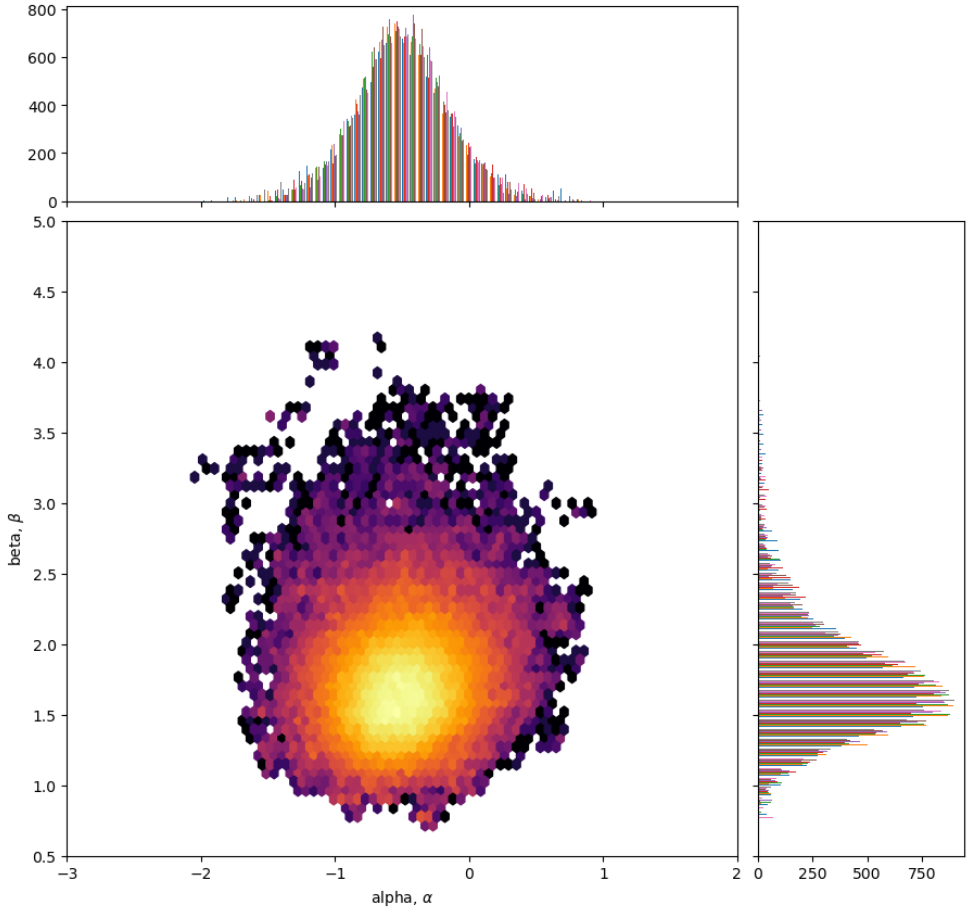

Figure 4: Visualisation of the joint posterior distribution for parameters α and β generated from a HMC algorithm with 8 independent chains, running for 10,000 steps with an adaptive step size using the NUTS kernel in TensorFlow. The contour plot highlights the density of the samples, the marginal histograms show the distribution of each parameter, and the overlaid traces represent the sampling paths of the Markov chains.

(i) Joint posterior of α and β:

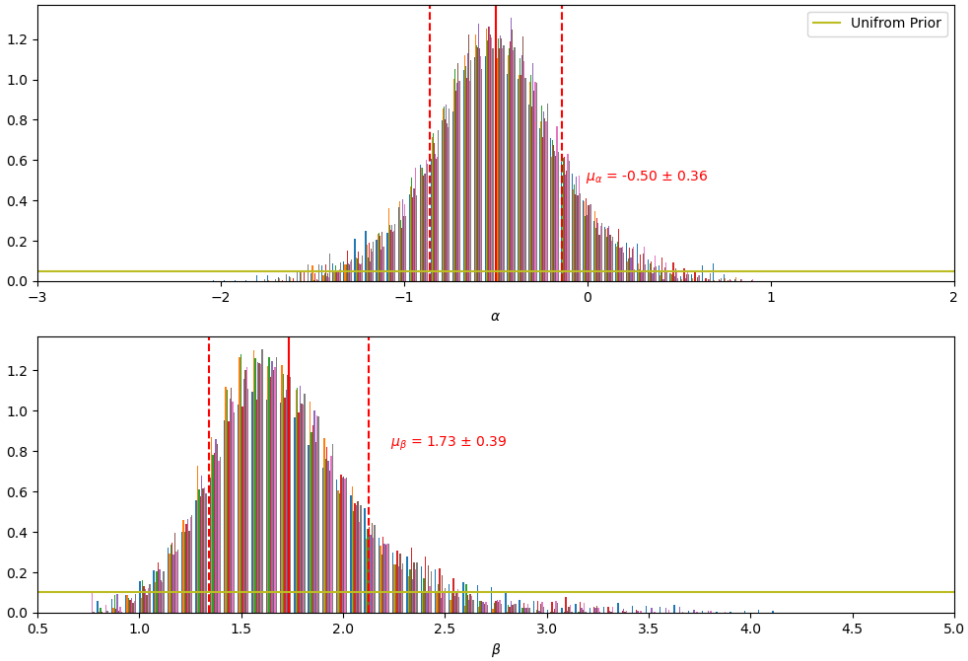

Figure 5: Marginal posterior distributions for parameters α and β from a HMC algorithm with 8 independent chains, running for 10,000 steps with an adaptive step size using the NUTS kernel in TensorFlow. The histograms represent the probability density of samples obtained for each parameter, with the solid line showing the mean μ and the dashed lines the standard deviations σ. The flat line represents the uniform prior. The plots reflect the degree of uncertainty in the estimation of each parameter, with α centered around -0.50 $\pm$ 0.36 and β around 1.73 $\pm$ 0.39.

(ii) Marginalised posterior of α and β:

(iii) Measurements of α and β: Alpha, α had a mean of -0.50 and standard deviation of 0.36. While beta, β had a mean of 1.73 and standard deviation of 0.39

(iv) Convergence diagnostic for Markov chain: Convergence was assessed using the trace plots, Gelmen-Rubin diagnostic, acceptance rate, the autocorrelation rate (τ) and the effective sample size (EFF).

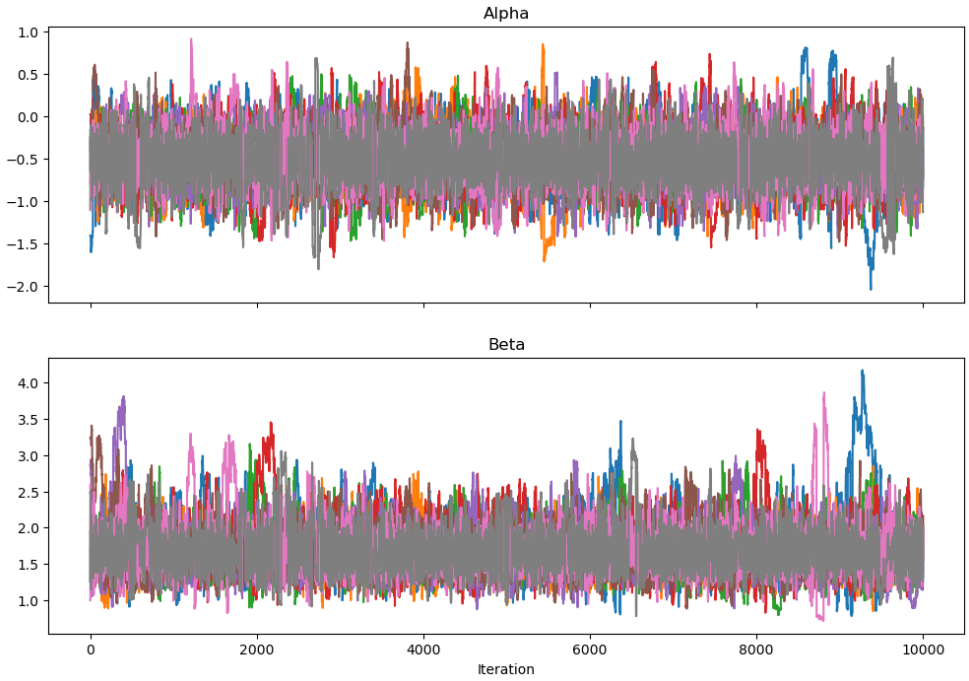

Figure 6: Trace plot displaying the sampling paths of 8 HMC chains for α and β across 10,000 iterations. Each colored line represents one of the multiple chains used in the analysis, illustrating the convergence behavior and mixing of the chains over the iterations. The stable and overlapping traces in the plots suggest good convergence and mixing for both parameters."

0.1.6 (vi) Selecting a Suitable Prior for I_0

Bibliography

[1] Duane, Kennedy, Pendleton & Roweth (1987) "Hybrid Monte Carlo", Physics Letters B, 195 (2) 216–222 doi:10.1016/0370-2693(87)91197-X.

[2] Radford Neal. MCMC Using Hamiltonian Dynamics. Handbook of Markov Chain Monte Carlo, 2011. https://arxiv.org/abs/1206.1901

[3] Hoffman & Gelman (2014) "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo", Journal of Machine Learning Research, 15 1593-1623,

[4] Abadi, Martín Abadi, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... others. (2016). Tensorflow: A system for large-scale machine learning. In 12th $USENIX$ Symposium on Operating Systems Design and Implementation ($OSDI$ 16).

0.2 Appendix

Since α and β are independent it makes sense to choose independent priors.

Alpha prior: There is no information about the location of the lighthouse, hence a non-informative uniform distribution over the range of the coast captures this,

$$p(x, y) = \begin{cases} (a-b)^{-1} & \text{for } a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Beta prior: Jefferys..?
