MPhil Data Intensive Science
University of Cambridge

Statistics for Data Science Coursework
The Lighthouse Problem

Andreas Vrikkis

March 28, 2024

LaTeX Word count: 2668

Contents

Introduction

In the lighthouse problem, a lighthouse is located somewhere off a straight coastline at position α along the shore and a distance β out to sea. It emits flashes of light at uniformly distributed random angles θ_k, where k is the flash number. The geometry of the problem is depicted in Fig.1. An array of detectors along the coastline record the locations of dected flashes, and later in the problem their intensities. In this report, we use this problem to demonstrate the use of Bayesian statistics to infer the location of the lighthouse from the observed flashes.

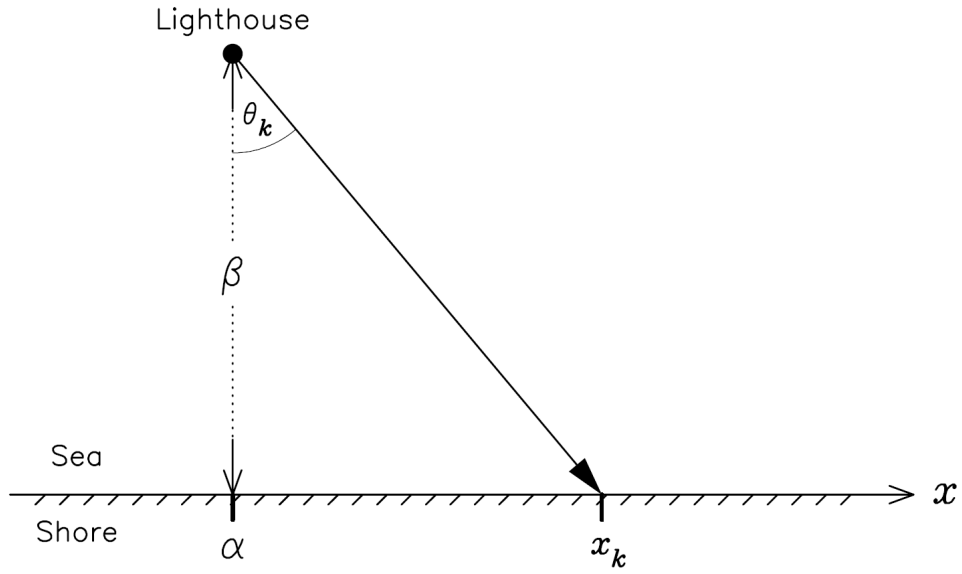

Figure 1: The geometry of the lighthouse problem, as depicted in ref.[1]. The x-axis is the shore, while the y-axis above the shore is the sea.

Part (i)

Referring to the Fig.1, we can see that the points (α, β), $(\alpha, 0)$ and (α, x_k) form a right angle triangle. We can relate the distances of this geometry and use elementary trigonometry to write the relation:

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

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

Part (ii)

The lighthouse rotates and emits flashes at uniformly-distributed random angle θ, but only the flashes that hit the shore are observed. Hence we restrict θ to satisfy the condition $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, and any angle in that region is equally likely. We can write the likelihood of θ as a uniform distribution in this region:

$$\mathcal{L}_x(\theta \mid \alpha, \beta) = \begin{cases} \frac{1}{\pi}, & \theta \in (-\frac{\pi}{2}, \frac{\pi}{2}) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

We now want to transform this probability density from being parameterised by angle θ to single flash location x. The probability distribution transformation rule [1] states that the

probability of x being in the small interval $[x, x+dx]$ is equal to the probability of θ being in the small interval $[\theta, \theta+d\theta]$. Using this, we can relate the probability density of θ, $p(\theta)$, to the probability density of x, $p(x)$, as:

$$p(x)dx = p(\theta)d\theta \tag{4}$$

Applying this to the likelihood we get:

$$\mathcal{L}_x(x \mid \alpha, \beta) = \mathcal{L}_x(\theta \mid \alpha, \beta) \left|\frac{d\theta}{dx}\right| \tag{5}$$

$$= \frac{1}{\pi} \left|\frac{d\theta}{dx}\right| \tag{6}$$

where in the last step we substituted the expression derived in eq.(3). Differentiating the relation in eq.(2) with respect to θ we get:

$$\frac{dx}{d\theta} = \beta \sec^2(\theta) \tag{7}$$

$$= \beta(1 + \tan^2(\theta)) \tag{8}$$

$$= \beta \left(1 + \left(\frac{x-\alpha}{\beta}\right)^2\right) \tag{9}$$

$$= \frac{(x-\alpha)^2 + \beta^2}{\beta} \tag{10}$$

$$\Rightarrow \frac{d\theta}{dx} = \frac{\beta}{\beta^2 + (x-\alpha)^2} \tag{11}$$

Finally we substitute the expression for $\frac{d\theta}{dx}$ into eq.(6) to obtain an expression of the likelihood with respect to x:

$$\mathcal{L}_x(x \mid \alpha, \beta) = \frac{1}{\pi} \left|\frac{d\theta}{dx}\right| \tag{12}$$

$$= \frac{\beta}{\pi(\beta^2 + (x-\alpha)^2)}. \tag{13}$$

Part (iii)

We begin by proving that the sample mean $\overline{x} = \frac{1}{N}\sum_{k=1}^{N} x_k$ cannot be a good estimator for α. First, we define the variable $z = (x-\alpha)/\beta$ and recast the likelihood function of eq.(13) in the form of the Cauchy distribution:

$$\mathcal{L}_x(x \mid \alpha, \beta) = \frac{\beta}{\pi\left(\beta^2 + (x-\alpha)^2\right)} \tag{14}$$

$$= \frac{1}{\beta} \frac{1}{\pi\left(1 + \left(\frac{x-\alpha}{\beta}\right)^2\right)} \tag{15}$$

$$= \frac{1}{\beta} \frac{1}{\pi\left(1 + z^2\right)} \tag{16}$$

The expectation of variable z is given by:

$$\mathbb{E}[z] = \frac{1}{\pi\beta}\int_{-\infty}^{\infty} z \cdot \frac{1}{(1+z^2)} dz \tag{17}$$

$$= \frac{2}{\pi\beta}\int_{0}^{\infty} z \cdot \frac{1}{(1+z^2)} dz \tag{18}$$

$$\tag{19}$$

where in the last step we used the symmetry of the integrand. Using integration by parts [2] (set $u = z$ and $v = \tan^{-1}(z)$), we get:

$$\mathbb{E}[z] = \left[z\tan^{-1}(z)\right]_0^{\infty} - \int_0^{\infty} \tan^{-1} z dz = \infty \tag{20}$$

The integral diverges and so the mean of the Cauchy distribution is undefined. This means that if we were to simulate the lighthouse problem many times and take the sample mean, the mean would never seettle down. The law of large numbers does not apply for $\overline{x}$ and it does not converge to a constant value. Thus, the sample mean cannot be a good estimator for α.

We can obtain an expression for the maximum likelihood estimate of α by differentiating the log-likelihood function with respect to α and setting it to zero. The likelihood for a set of N flashes $\{x\}$ is given by:

$$\mathcal{L}_x(\{x_k\} \mid \alpha, \beta) = \prod_{k=1}^{N} \frac{\beta}{\pi(\beta^2 + (x_k - \alpha)^2)} \tag{21}$$

Taking the logarithm of the likelihood, we get:

$$\log \mathcal{L}_x(\{x_k\} \mid \alpha, \beta) = \sum_{k=1}^{N}\left[\log\left(\frac{\beta}{\pi}\right) - \log\left[\beta^2 + (x_k - \alpha)^2\right]\right] \tag{22}$$

$$= N\log\left(\frac{\beta}{\pi}\right) - \sum_{k=1}^{N}\log\left[\beta^2 + (x_k - \alpha)^2\right] \tag{23}$$

Differentiating the log-likelihood with respect to α gives:

$$\frac{\partial \log \mathcal{L}_x\left(\{x_k\} \mid \alpha, \beta\right)}{\partial\alpha} = \sum_{k=1}^{N}\frac{\partial}{\partial\alpha}\left[-\sum_k \log\left(\beta^2 + (x_k - \alpha)^2\right)\right] \tag{24}$$

$$= \sum_{k=1}^{N}\frac{2\left(x_k - \alpha\right)}{\beta^2 + \left(x_k - \alpha\right)^2} \tag{25}$$

which we set to zero to obtain the Maximum Likelihood Expression of α, $\hat{\alpha}$:

$$\sum_{k=1}^{N}\frac{x_k - \hat{\alpha}}{\beta^2 + \left(x_k - \hat{\alpha}\right)^2} = 0 \tag{26}$$

This is a polynomial equation of degree $2N - 1$ which does not have a closed-form solution for $N = 20$ flashes (our dataset in this problem). It has been shown by the authors of ref.[2] that multiple roots exist for this equation, and the maximum likelihood estimate of α is not necessarily unique. The frequentist colleague's claim that the most likely location for a flash is at $x = \alpha$ is correct for the case of solving eq.(26) for $N = 1$. However, it is not true for $N > 1$, and the sample mean clearly does not satisfy the maximum likelihood equation.

Part (iv)

We assign a simple uniform prior probability distribution to α in the form:

$$\pi(\alpha) = \begin{cases} \frac{1}{|\alpha_{\max} - \alpha_{\min}|} & \alpha_{\min} \leqslant \alpha \leqslant \alpha_{\max}, \\ 0 & \text{otherwise}, \end{cases} \tag{27}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are the limits of the coastline (or the region covered by the photodectors). This is because we are given no information about the lighthouse location, and so we assume that the lighthouse is equally likely to be anywhere along the coastline. Also, α defines a location and so must be invariant under translations of the origin by a small constant $\Delta\alpha$. This prior satisfies this condition as:

$$\pi(\alpha)d\alpha = \pi(\alpha + \Delta\alpha)d(\alpha + \Delta\alpha) \tag{28}$$

For the same reasons we set the prior of β to be uniform, given by:

$$\pi(\beta) = \begin{cases} \frac{1}{\beta_{\max}} & 0 \leqslant \beta \leqslant \beta_{\max}, \\ 0 & \text{otherwise}, \end{cases} \tag{29}$$

where $\beta_{\max}$ is the maximum possible value of β and the lighthouse is not located at the shore, thus it is not possible for β to be negative. We discuss how we chose the prior limits in the next section.

Part (v)

This part is presented by first expressing the posterior distribution for the lighthouse problem. We then describe the Markov Chain Monte Carlo (MCMC) method used to sample from the posterior distribution. The methodology and convergence diagnostics follow, and we conclude with the results obtained from the MCMC algorithm.

Posterior Distribution

We now turn our attention to drawing stochastic samples from the posterior distribution $P(\alpha, \beta \mid \{x_k\})$ where $\{x_k\}$ is the set of observed flashes. Since all flashes are emitted at independent angles θ_k, the likelihood function for the set of flashes is given by the product of the likelihood functions for each flash:

$$\mathcal{L}_x(\{x_k\} \mid \alpha, \beta) = \prod_{k=1}^{N} \mathcal{L}_x(x_k \mid \alpha, \beta) \tag{30}$$

where N is the number of observed flashes and x_k is the location of the k-th flash. We can now write the joint posterior distribution of α and β as:

$$P(\alpha, \beta \mid \{x_k\}) \propto \mathcal{L}_x(\{x_k\} \mid \alpha, \beta)\pi(\alpha)\pi(\beta) \tag{31}$$

where $\pi(\alpha)$ and $\pi(\beta)$ are the prior distributions of α and β respectively. In this formulation we have omitted the normalisation constant (evidence) Z, which is the integral of the product of the likelihood and the prior over the entire parameter space. In the context of MCMC sampling, the focus is on the relative probabilities of different points within the parameter space. This means that the evidence Z becomes irrelevant to our calculations, as it cancels out when comparing probabilities of different parameter values. It is for the same reason that we

drop the constant priors $\pi(\alpha)$ and $\pi(\beta)$ as defined in eq.(27) and eq.(29) respectively. Since these priors are constants, they do not influence the relative probabilities of different points in the parameter space.

In the course of sampling, we utilise the logarithm of the posterior distribution. This approach is justified because the logarithm is a monotonic function, which ensures that the relative ordering of probabilities in the parameter space is maintained. Regions of higher probability in the original posterior are still higher in the log-transformed space, while being more numerically stable. We can thus express the logarithm of the posterior distribution as:

$$\log P(\alpha, \beta \mid \{x_k\}) = \log \mathcal{L}_x(\{x_k\} \mid \alpha, \beta) \tag{32}$$

$$\propto N \log\left(\frac{\beta}{\pi}\right) \sum_{k=1}^{N} \log\left(\beta^2 + (x_k - \alpha)^2\right) \tag{33}$$

MCMC Sampling Method

We briefly describe the MCMC sampling method used to sample from the posterior distribution. We chose to use the package `zeus` which is a Python implementation of Ensemble Slice Sampling (ESS) [3]. This package is a well-tested and efficient sampler, which requires minimal hand-tuning and does not require gradients. Slice sampling operates on the concept that sampling from a distribution with density $P(x)$ is equivalent to sampling uniformly from the area under the curve of $f(x)$, where $f(x) \propto P(x)$. This means that our posterior distribution $P(\alpha, \beta \mid \{x_k\})$ does not need to be normalised. The method involved an additional variable y (referred to as 'height') to create a join distribution $P(x, y)$ that is uniform over the region under the curve of $f(x)$. To sample from the marginal distribution $P(x)$, we sample y uniformly from the interval $[0, f(x)]$ and then sample x from the conditional distribution $P(x \mid y)$. Finally we marginalise out y to obtain the samples from $P(x)$. A high level overview of the algorithm for a simple univariate case is shown in Algorithm 1.

Algorithm 1 Univariate Slice Sampling

1: **Input:** $f(\alpha), \alpha_0,$ Number of steps n ▷ Initial guess for $\alpha = \alpha_0$
2: **for** $i = 0, 1, \ldots, n$ **do** ▷ Iterate over the number of steps
3: $y_i \sim \text{Uniform}(0, f(\alpha_i))$ ▷ Sample 'height auxialiary variable
4: $\alpha_{i+1} \sim \text{Uniform}(\{\alpha : f(\alpha) > y_i\})$ ▷ Sample uniformly from the slice
5: **end for**

The challenging part of this algorithm is constructing the slice interval in step 4. Ref.[4] proposed to use the 'stepping out' and 'shrinking' procedure that uses a length scale free parameter μ to construct the slice vector. `zeus` adapts this idea using ESS, which is a stochastic optimisation scheme to adaptively tune the length scale parameter μ to the local geometry of the posterior distribution. To generalise this procedure to multidimensional parameter spaces, `zeus` initialises multiple parallel chains, called walkers, which evolve independently and exchange information through the ensemble. Their positions carry information about the correlations between parameters, which are used to construct the slice vectors in parameter space to sample from. The complete algorithm behind `zeus`'s ESS contains multiple subtleties and can be found in ref.[5].

Chain Initialisation & Convergence Diagnostics

Based on empirical evidence, the author of Zeus suggests that initialising walkers near the Maximum A Posteriori (MAP) estimate can reduce the burn-in period significantly [3]. Well-spread out points enhances exploration of the parameter space. We identify optimal walker starting points by plotting a mesh of points over the parameter space of the unnormalized log posterior (eq. 33). We convert the outputs to exponentials, and plot a contour heatmap for regions encompassing the MAP estimate (Fig. 2). From this, we select the region $-2 < \alpha < 1$ and $0 < \beta < 3$ for uniformly distributing the walkers' initial positions. The MCMC algorithm is then executed for 10000 steps with 10 walkers, in line with the recommendation of having more than 2-3 times the number of parameters as the number of walkers [3]. We initiate two independent samplers using `zeus.EnsembleSampler()` for a total of 10000 steps each. The second run is used for convergence diagnostics.

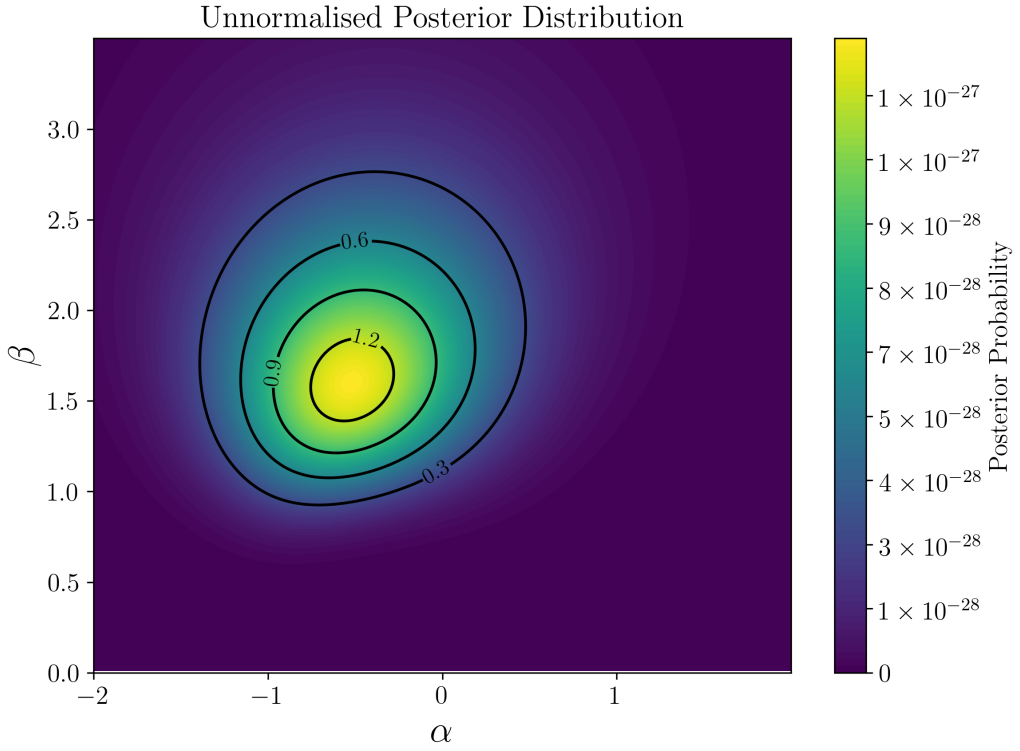

Figure 2: The contours of the logarithm of the posterior distribution of α and β.

After acquiring samples from the MCMC algorithm, we address sample correlation by computing the Integrated Autocorrelation Time (IAT). IAT reflects the step interval between any two samples in the chain required to ensure that they are uncorrelated. We employ `zeus`'s implementation (`zeus.AutoCorrTime()`) which uses a fast Fourier Transform method to estimate the IAT per parameter. The maximum IAT τ of all the parameters is used to thin the chain by rounding up to the nearest integer and only keeping every τ-th sample.

To check for convergence, we use the (Split-)Gelman-Rubin statistic $\hat{R}$. This statistic assesses mixing and stationarity by splitting multiple independent MCMC chains and calculating the ratio of between-chain and within-chain variances. Consider M parallel markov chains, x^m, each of which has N samples. The mth chain is given by $x_n^{(m)}$, where n is the iteration number

and m is the chain number. The between-chain variance B is given by:

$$B = \frac{N}{M-1} \sum_{m=1}^{M} \left(\bar{x}^{(m)} - \overline{x_*}\right)^2 \tag{34}$$

$$\tag{35}$$

where

$$\bar{x}^{(m)} = \frac{1}{N} \sum_{n=1}^{N} x_n^{(m)}, \text{ mean value of chain } m \tag{36}$$

$$\overline{x_*} = \frac{1}{M} \sum_{m=1}^{M} \bar{x}^{(m)}, \text{ mean of the means of all chains} \tag{37}$$

The within-chain variance W in then given by:

$$W = \frac{1}{M} \sum_{m=1}^{M} s_m^2 \tag{38}$$

where

$$s_m^2 = \frac{1}{N-1} \sum_{n=1}^{N} \left(x_n^{(m)} - \bar{x}^{(m)}\right)^2, \text{variance chain } m \tag{39}$$

The $\hat{R}$ statistic is then given by:

$$\hat{R} = \sqrt{\frac{N-1}{N} + \frac{1}{N}\frac{B}{W}}. \tag{40}$$

We repeat this process for each parameter. For convergence, $\hat{R}$ should approach 1, with a typical cutoff of 1.1.

We implement a function to compute the Split-R Gelman-Rubin statistic for the two indepented chains initialised, splitting each into two halves and treating them as four independent chains. This split helps assess stationarity within each chain. We apply this process to all 100-iteration segments of the chain, obtaining the $\hat{R}$ statistic as a function of the number of iterations.

Results

Fig.3 displays the paths of two walkers from a single ensemble during the initial 500 steps. All the starting points are within the defined interval that was based on the posterior contours. It is difficult to discern convergence from these plot (we include them for illustrative purposes). The $\hat{R}$ statistic over all iterations is shown in Fig.4, calculated as described in the previous section. Note that the total number of iterations is 100000 because we have 10 walkers and 10000 steps. Both $\hat{R}(\alpha)$ and $\hat{R}(\beta)$ fall below the threshold value of 1.1 after around 500 iterations. The chains settle into a stable value after around 2000 iterations. While we could discard very few interations as the burnin, we conservatively discard the first 2000 iterations.

The IAT values obtained were $\tau_\alpha = 3.19$ and $\tau_\beta = 4.77$. We take the larger of the two values as the maximum autocorrelation time, which is $\tau = 4.77$. Based on this, we thin the chains by a factor of 5 (by rounding up τ) to obtain the final samples. Discarding the burnin and thinning us with 19600 samples for each parameter.

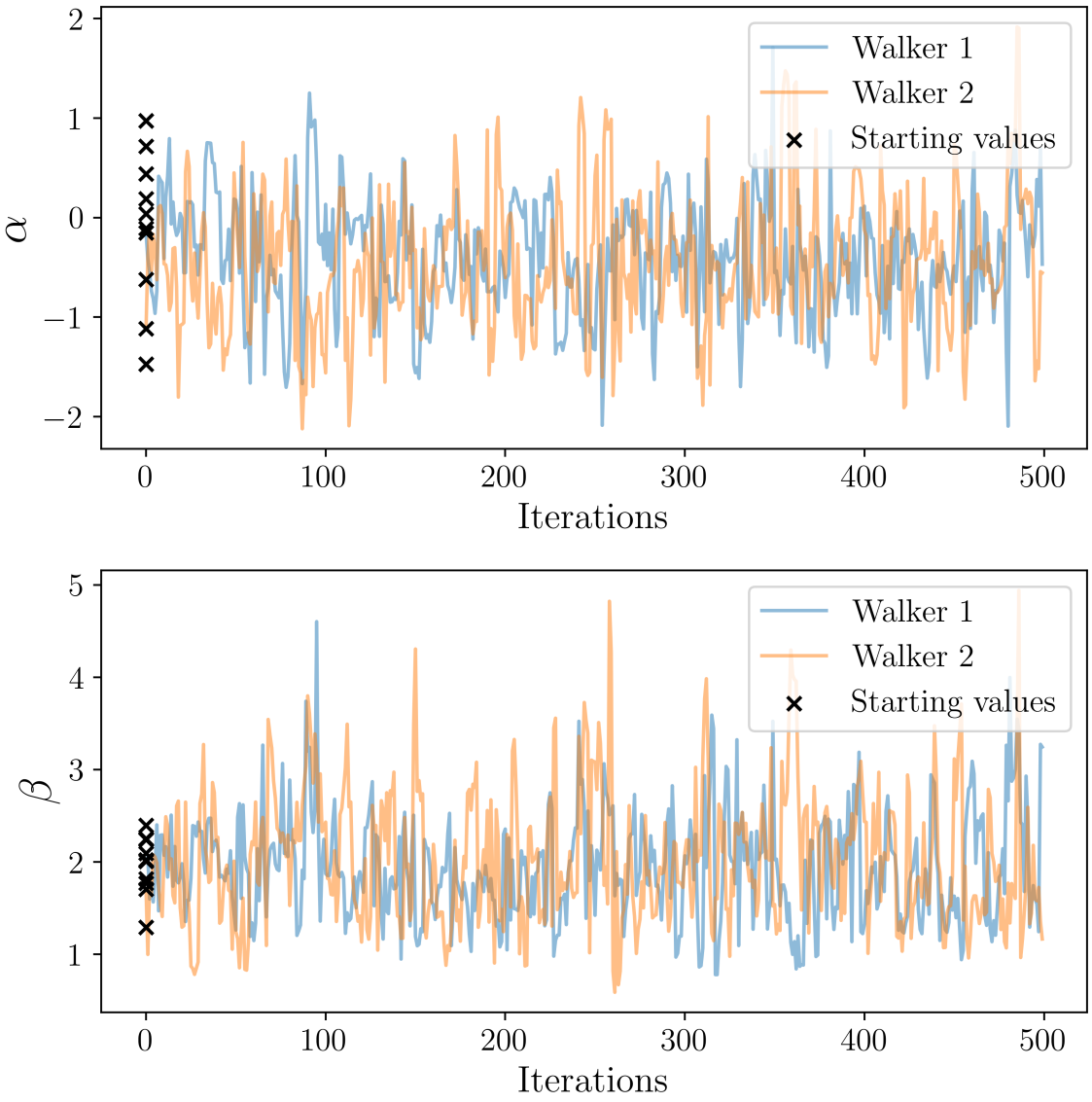

Figure 3: The first two walkers for the first 500 steps of the MCMC algorithm. The plot at the top shows the walkers in the parameter space of α and the bottom plot shows the walkers in the parameter space of β. The starting of all 10 walkers are shown as black crosses. Only the first two walkers and the first 500 iterations out of 100000 are shown for clarity.

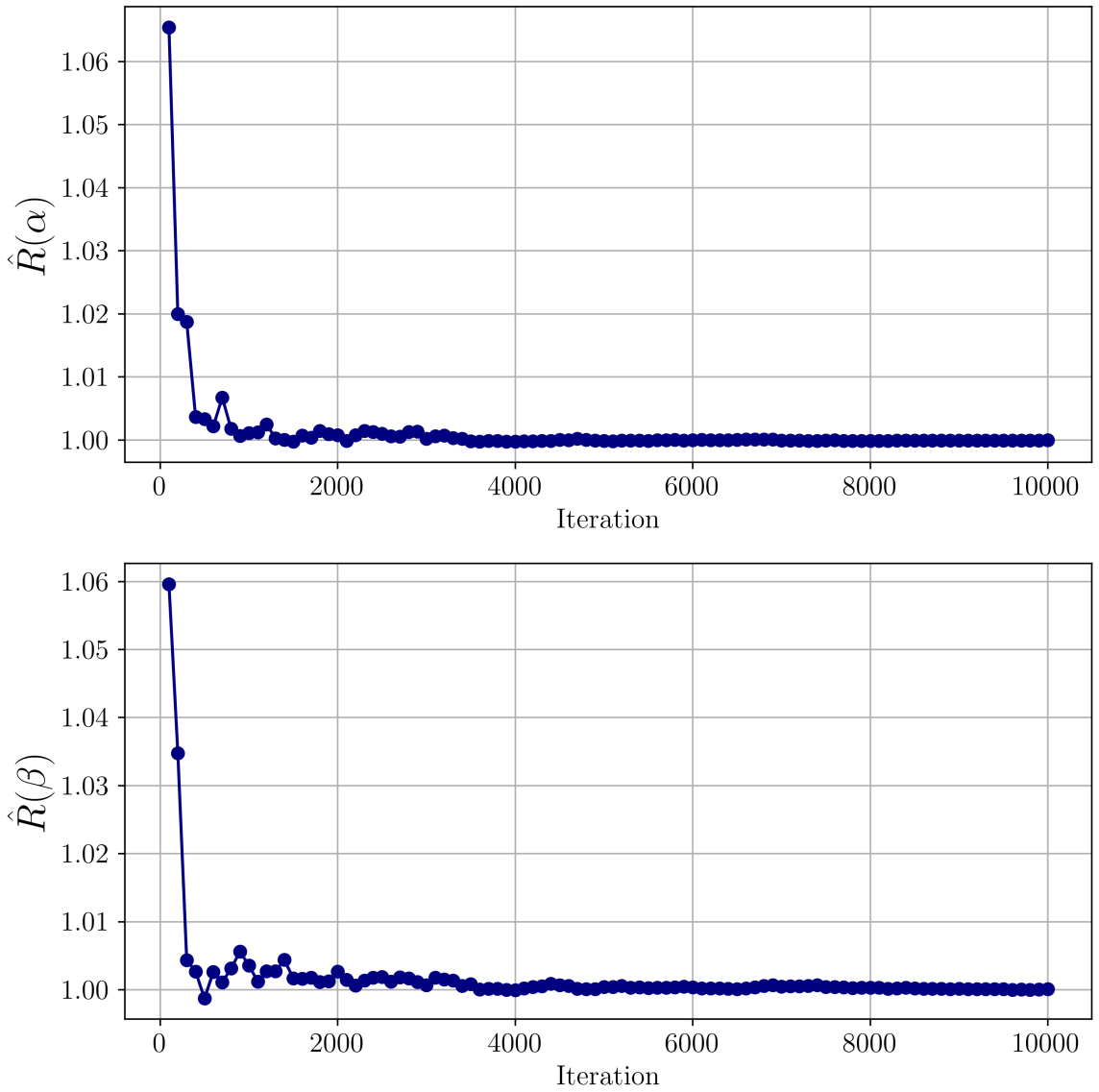

Figure 4: The Gelman-Rubin statistic as a function of the number of iterations for the parameters α and β. The statistic is calculated for the two independent chains obtained from the ensemble samplers. The statistic falls below the threshold value of 1.1 after around 500 iterations.

Taking the mean and standard deviation of the samples, we obtain the following values:

$$\alpha = -0.45 \pm 0.60 \tag{41}$$

$$\beta = 1.96 \pm 0.67. \tag{42}$$

Because we thinned the chains by the autocorrelation time, this estimate takes into account the correlation between the samples. The corner plot of the samples containing histograms of the samples and the 2D joint distribution is shown in Fig.5.

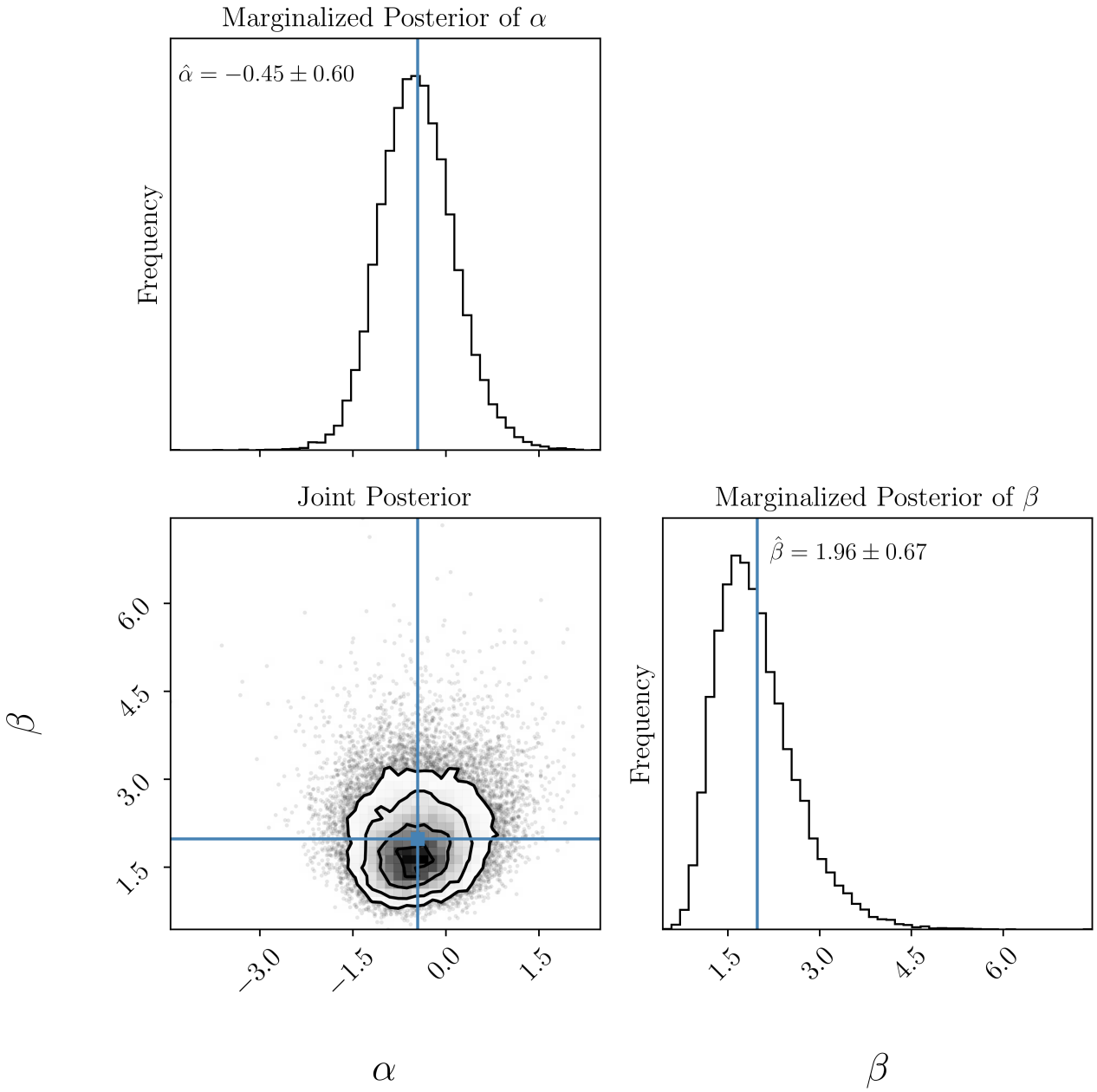

Figure 5: Corner plot of the MCMC samples from the posterior distribution of α and β. The bottom left plot shows the 2D histogram of the samples, which transitions to a scatter plot in the low-density regions. This forms the joint posterior distribution and is marked by the blue lines where the mean of the samples is found. The top and right plots show the 1D histograms of the samples for α and β respectively. The blue lines show the mean of the samples.

Part (vi)

We now consider the case where the intensities of the flashes I_k are also recorded. We aim to infer the absolute intensity I_0 of the lighthouse, as well as obtain better estimates of its location. The method employed is largely similar to the previous case, with the likelihood function being modified to account for the intensity of the flashes. In the following sections, we highlight the differences in our methodology, and present the results.

To choose a suitable prior for the intensity I_0, we consider the fact that it is a positive scale parameter. This means that the prior must be invariant under a change of a constant factor Δ

(e.g change in units). This means:

$$\pi(I_0)dI_0 = \pi(\Delta I_0)d(\Delta I_0) \tag{43}$$

A suitable choice is the Jeffrey's prior, which is invariant under a multiplicative constant rescaling. We set a log-uniform prior for I_0:

$$\pi(I_0) = \begin{cases} \frac{1}{I_0 log(I_{0max}/I_{0min})}, & I_{0min} \leqslant I_0 \leqslant I_{0max} \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

We choose $I_{0min} = 0.1$ and $I_{0max} = 10$ as the limits of the prior, which cover the range of the provided intensities I_k with added margin.

Part (vii)

Posterior Distribution with Intensity

The likelihood for each intensity measurement I is given by:

$$\mathcal{L}_I\left(\log I \mid \alpha, \beta, I_0\right) = \frac{\exp\left(\frac{-(\log I - \mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}} \tag{45}$$

where $\mu = \log(I_0/d^2)$ is the expected log-intensity, $d^2 = \beta^2 + (x-a)^2$ is the light house-detector distance, and $\sigma = 1$. The log-likelihood of a single intensity measurement is then given by:

$$\log \mathcal{L}_I\left(\log I \mid \alpha, \beta, I_0\right) = -\frac{(\log I - \mu)^2}{2\sigma^2} - \log\left(\sqrt{2\pi}\sigma\right) \tag{46}$$

The likelihood for the set of intensities $\{I_k\}$ is given by the product of the likelihoods for each intensity measurement:

$$\mathcal{L}_I\left(\{I_k\} \mid \alpha, \beta, I_0\right) = \prod_{k=1}^{N} \mathcal{L}_I\left(\log I_k \mid \alpha, \beta, I_0\right) \tag{47}$$

The log-likelihood for the set of intensities is then given by the sum of the log-likelihoods for each intensity measurement:

$$\log \mathcal{L}_I\left(\{I_k\} \mid \alpha, \beta, I_0\right) = -\sum_{k=1}^{N} \frac{(\log I_k - \mu)^2}{2\sigma^2} - N \log\left(\sqrt{2\pi}\sigma\right) \tag{48}$$

Since the location and intensity measurements are independent, the likelihood for a combined location and intensity measurement is the product of the likelihoods:

$$\mathcal{L}\left(\{x_k, I_k\} \mid \alpha, \beta, I_0\right) = \mathcal{L}_x\left(\{x_k\} \mid \alpha, \beta\right) \mathcal{L}_I\left(\{I_k\} \mid \alpha, \beta, I_0\right), \tag{49}$$

and so their combined log-likelihood is just their sum:

$$\log \mathcal{L}\left(\{x_k, I_k\} \mid \alpha, \beta, I_0\right) = \log \mathcal{L}_x\left(\{x_k\} \mid \alpha, \beta\right) + \log \mathcal{L}_I\left(\{I_k\} \mid \alpha, \beta, I_0\right) \tag{50}$$

To implement this combined log-likelihood in the MCMC sampler, we use the same function implementation for the location log-likelihood as before, and simply add to it the log-likelihood

of eq.(48). To complete the combined log-posterior, we add the log-prior of eq.(44) to the combined log-likelihood, to give:

$$\log P\left(\alpha, \beta, I_0 \mid \{x_k, I_k\}\right) \propto \log \mathcal{L}\left(\{x_k, I_k\} \mid \alpha, \beta, I_0\right) + \log \pi(I_0) \tag{51}$$

We again drop the constant priors for α and β as they do not influence the relative probabilities of different points in the parameter space.

For the walker starting points, we used the same range for α and β as in the previous case, in which we choose the starting points randomly from a uniform distribution. For I_0, we fix the values of α and β to the previously obtained mean values $\hat{alpha} = -0.45, \hat{\beta} = 1.96$, and plot a series of points over the unnormilised log posterior. This allows to gauge a sensible range for I_0, and is shown in Fig.6. Based on this plot, we choose the starting points for I_0 to be in the range $2 < I_0 < 8$. As before, we run two independent samplers for 10000 steps each, and use the second run for convergence diagnostics. 10 walkers are used for each run.

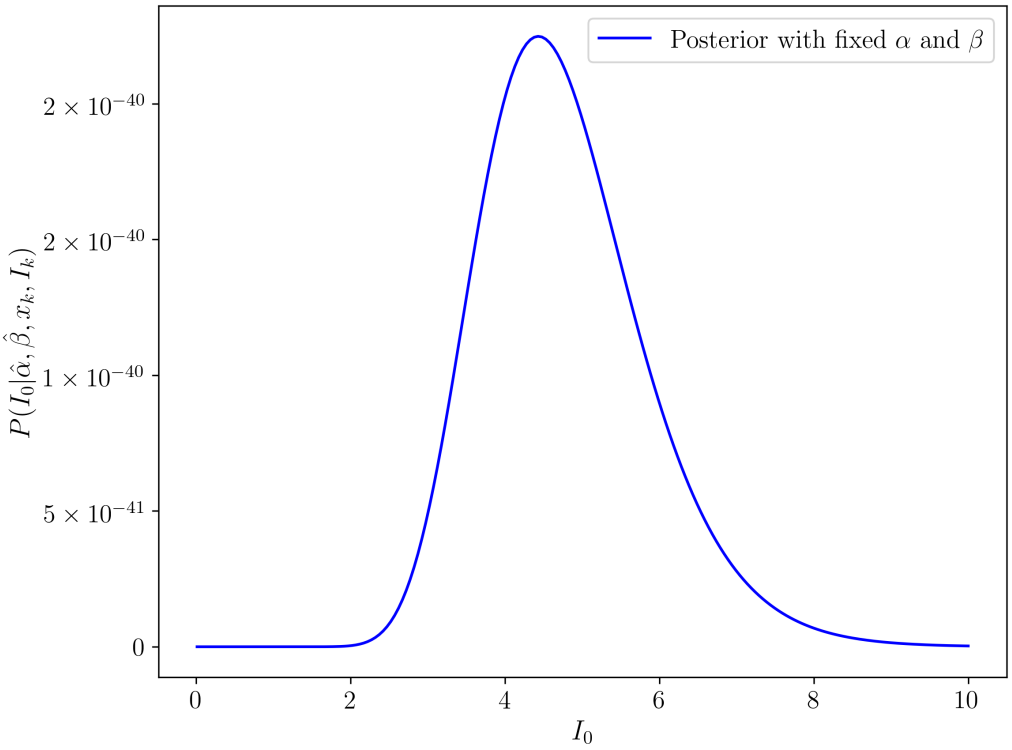

Figure 6: 1-dimensional slice of the posterior $P(I_0 \mid \alpha = -0.45, \beta = 1.96, \{x_k\}, \{I_k\})$. This allows to obtain the MAP estimate of I_0 and choose a suitable range for the starting positions.

Results

Fig.7 displays the paths of three walkers from a single ensemble during the initial 500 steps. We can see that it takes only a few steps for the walkers to start converging into a smaller region of the parameter space. The $\hat{R}$ statistic over all iterations is shown in Fig.8, calculated as described in the previous section. All three parameters α, β, and I_0 fall below the threshold value of 1.1 after around 1000 iterations. The chains settle into a stable value after around 6000 iterations. Again, we conservatively discard the first 2000 iterations as the burnin.

The IAT values obtained were $\tau_\alpha = 5.41$, $\tau_\beta = 7.70$, and $\tau_{I_0} = 8.21$. We take the largest of the three values as the maximum autocorrelation time, which is $\tau = 8.21$. Based on this, we thin the chains by a factor of 9 (by rounding up τ) to obtain the final samples. Discarding the burnin and thinning us with 10444 samples for each parameter. The IAT is higher in this case

because the intensity measurements are correlated with the location measurements, and so the chains take longer to decorrelate.

Taking the mean and standard deviation of the samples, we obtain the following values:

$$\alpha = -0.197 \pm 0.316 \tag{52}$$

$$\beta = 1.514 \pm 0.361 \tag{53}$$

$$I_0 = 3.74 \pm 1.217. \tag{54}$$

The corner plot of the samples containing the 1-d marginal distributions and the 2-d joint distributions of the parameters is shown in Fig.9. On this plot, each parameter is quoted with its median and $1 - \sigma$ quantiles. We can also see that I_0 and β are positively correlated, which is expected as the intensity of the flashes is proportional to the inverse square of the distance from the lighthouse.

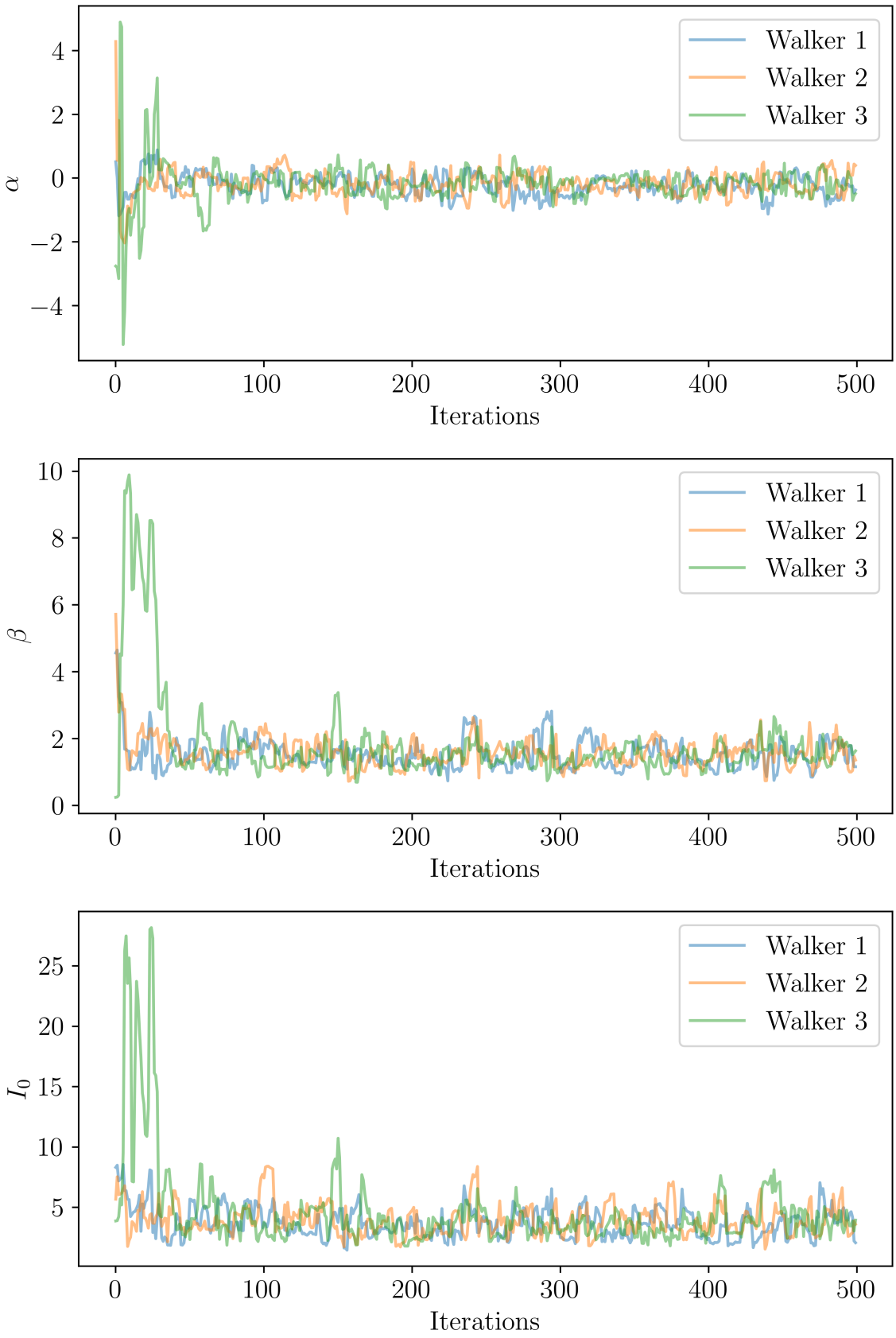

Figure 7: The first three walkers for the first 500 steps of the MCMC algorithm. The plot at the top shows the walkers in the parameter space of α, the middle plot shows the walkers in the parameter space of β, and the bottom plot shows the walkers in the parameter space of I_0. Only the first three walkers and the first 500 iterations out of 10000 are shown for clarity.

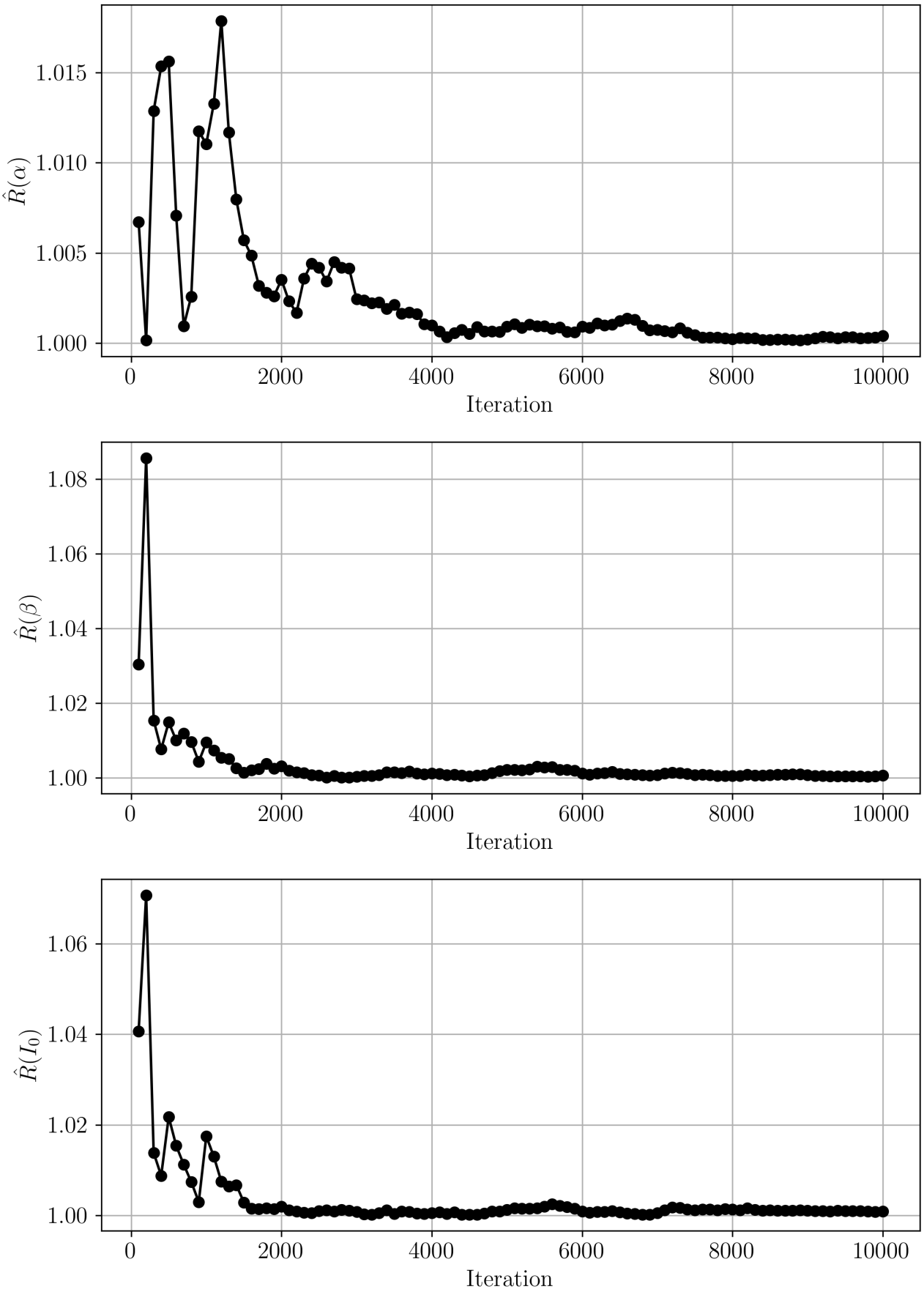

Figure 8: The Gelman-Rubin statistic as a function of the number of iterations for the parameters α, β, and I_0. The statistic is calculated for the two independent chains obtained from the ensemble samplers. The statistic falls below the threshold value of 1.1 after around 500 iterations.

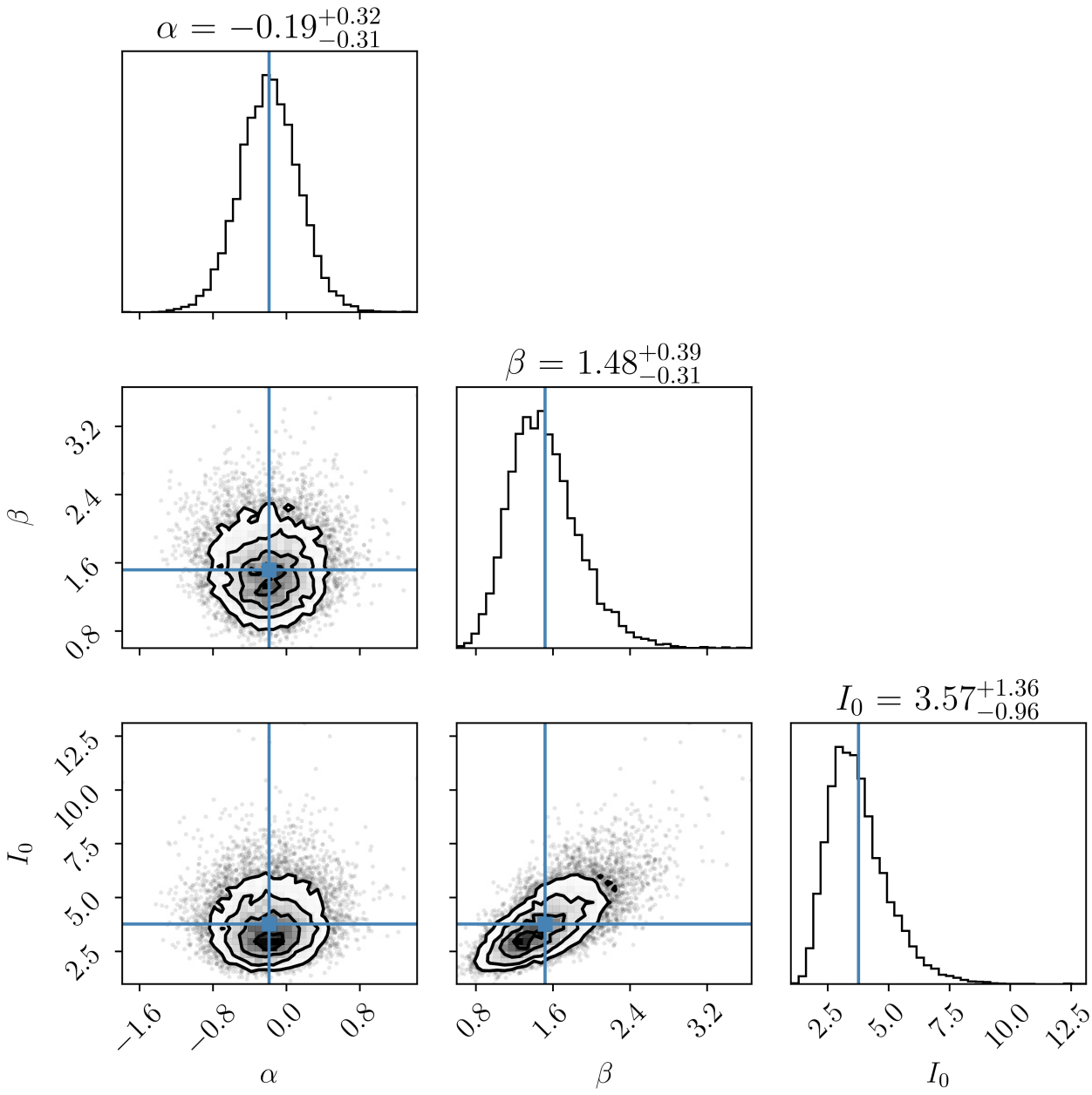

Figure 9: Corner plot of the MCMC samples from the posterior distribution of α, β, and I_0. The bottom left plot shows the 2D histogram of the samples, which transitions to a scatter plot in the low-density regions. This forms the joint posterior distribution and is marked by the blue lines where the mean of the samples is found. The outer plots show the 1D marginalised histograms of the samples for α, β, and I_0 respectively. The blue lines show the mean of the samples. The parameter medians are quoted with their $1-\sigma$ quantiles.

Part (viii)

When including the intensity data in the analysis, the standard deviations of α and β reduce by 47.3% and 46.1% respectively. This is because the intensity data provides additional information about the location of the lighthouse, which reduces the uncertainty in the location estimates. Hence, their addition improves the estimates of α and β.

References

[1] D. S. Sivia. *Data Analysis: A Bayesian Tutorial*. Oxford University Press, June 2006.

[2] James A. Reeds. Asymptotic Number of Roots of Cauchy Location Likelihood Equations. *The Annals of Statistics*, 13(2):775–784, June 1985. Publisher: Institute of Mathematical Statistics.

[3] Minas Karamanis, Florian Beutler, and John A. Peacock. zeus: A Python implementation of Ensemble Slice Sampling for efficient Bayesian parameter inference. *Monthly Notices of the Royal Astronomical Society*, 508(3):3589–3603, October 2021. arXiv:2105.03468 [astro-ph, physics:physics].

[4] Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, June 2003. Publisher: Institute of Mathematical Statistics.

[5] Minas Karamanis and Florian Beutler. Ensemble Slice Sampling: Parallel, black-box and gradient-free inference for correlated & multimodal distributions. *Statistics and Computing*, 31(5):61, September 2021. arXiv:2002.06212 [astro-ph, stat].

Appendix

Autogeneration Tools

Co-pilot was used in the following ways:

- Autocompleting documentation for functions and scripts.
- Sometimes employing the autocomplete feature for standard script routines (though these often require corrections and serve more as templates), such as:
  - #plotting the data: Allowed to autocomplete the plotting script.
- Understanding error messages with the "Explain this error" button next to the error message.

General Questions

Question: How can I change a function name and have it change throughout my workspace?

Answer: In Visual Studio Code, you can use the "Rename Symbol" feature to change a function name across your workspace. Here are the steps:

1. Place your cursor on the function name you wish to change.
2. Right-click the function name, and select "Rename Symbol" from the context menu. Alternatively, you can use the shortcut F2.
3. Enter the new function name and press Enter.

VS Code will automatically find and update all occurrences of the function name in your workspace, including the function definition and all locations where the function is called.