Statistical Computing Assignment 3

Mahmood-Ali Parker (PRKMAH005)

07/04/2021

Contents

(a) Using the Acceptance-rejection Algorithm to Sample from $\pi(\alpha, \beta | x)$

Understanding the Posterior Distribution

We need to sample from our posterior distribution, $\pi(\alpha, \beta | x)$. We know that the posterior distribution is given by

$$\pi(\alpha, \beta | x) \propto \pi(\alpha, \beta) * L(x, \alpha, \beta)$$

and we're given the likelihood, $L(x, \alpha, \beta)$ as

$$L(x, \alpha, \beta) \propto (\frac{\beta}{A(\alpha, \beta)})^{200} * \Pi_{i=1}^{200}(\frac{1}{\beta^2 + (x_i - \alpha)^2}), \beta > 0, 0 < \alpha < 5$$

We also know that α and β are independent and hence we can deduce that:

$$\pi(\alpha, \beta) = \pi(\alpha)\pi(\beta)$$

Given that $\alpha{\sim}$U(0,5) and $\beta{\sim}\Gamma$(4,8) we have our posterior distribution as:

$$\pi(\alpha, \beta | x) \propto \beta^3 e^{-8\beta}(\frac{\beta}{A(\alpha, \beta)})^{200} * \Pi_{i=1}^{200}(\frac{1}{\beta^2 + (x_i - \alpha)^2})$$

For ease of computation and to prevent numerical overflow and underflow, we can also derive the log posterior distribution:

$$\log(\pi(\alpha, \beta | x)) \propto 3 \log \beta - 8\beta + 200 \log(\frac{\beta}{A(\alpha, \beta)}) + \Sigma_{i=1}^{200} \log(\frac{1}{\beta^2 + (x_i - \alpha)^2})$$

We start by creating a function in R to compute the vector of log-posterior values of a parameterised (α, β) vector pair given the data, X according to the formula above.

Using, this log-posterior function, we create another function to compute the exponent of the log-posterior to get the posterior values of a parameterised (α, β) vector pair.

With the posterior and log-posterior functions, we can now plot these functions using R's contour() and persp() functions.
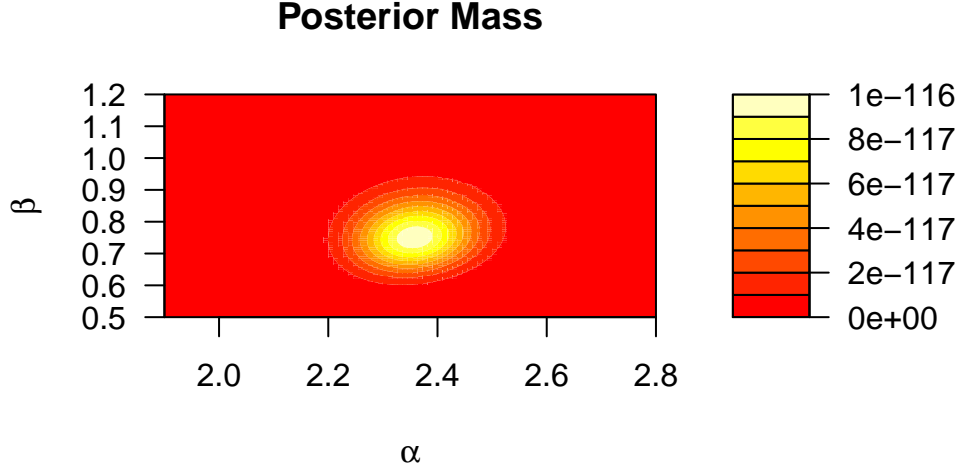
Posterior Mass

Figure 1: *Contour plot of the posterior mass of α ϵ [1.9, 2.8] and β ϵ [0.5, 1]. Plotted with R's filled.contour()*

From Figure 1, we can observe that the mode of the distribution is at approximately $\alpha = 2.35$ and $\beta = 0.75$ and that the majority of the mass lies between $\alpha\epsilon[2.2, 2.5]$ and $\beta\epsilon[0.6, 9]$. To get exact modal values for the parameters, we will optimise the log-posterior distribution to find α and β. The 3-D distribution plots in Figure 5 (see appendix) also show that a large mass of the posterior distribution is in a small area (denoted by a sharp peak and flat everywhere else).

Finding the Mode of the Posterior Distribution

We start the sampling process by finding the modal values for α and β. I implemented the Gauss-Seidel algorithm with a function, gaussSeidel(a,b,tol,fn), by iterating over R's optimize() function with a while-loop until both α and β converge to optimal parameter values (subject to a tolerance). The function returns the optimal α, β and the value of the objective function at convergence. Also worth noting is that the search ranges are set to $\alpha\epsilon[0, 5]$ and $\beta\epsilon[0, 10^{15}]$.

Running the gaussSeidel() function with (arbitrary) starting points of (3,10) to optimise the log-posterior function yields $\alpha = 2.358$ and $\beta = 0.75$ with a log-posterior objective value of -267.111. We can therefore conclude that the mode of our posterior distribution is the point (2.358, 0.75). We will use these modal parameters to help with the sampling process.

Accept-reject Sampling

We can now proceed with the Accept-reject method. The first step is to define our target distribution as the posterior distribution, $\pi(\alpha, \beta|x)$. This is the distribution we intend to obtain a sample from.

Second, we define the candidate distribution, $h(\alpha, \beta)$. The candidate distribution is the distribution that we will sample from for the acceptance-rejection algorithm until we have a sufficient number of samples. We choose our candidate distribution, $h(\alpha, \beta)$ to be our prior distribution $\pi(\alpha, \beta)$:

$$h(\alpha, \beta) = \pi(\alpha, \beta) \propto \beta^3 e^{-8\beta}$$

We choose this because it is easy to sample from the $\Gamma(4, 8)$ and U(0,5) distributions and, importantly, with our constant C, $Ch(\alpha, \beta)$ is close to our target distribution.

C is defined as follows:

$$C = \arg\max_{\alpha,\beta} \frac{\pi(\alpha,\beta)L(x,\alpha,\beta)}{h(\alpha,\beta)}$$

Since the prior and candidate distributions are the same we can simplify this to:

$$C = \arg\max_{\alpha,\beta} L(x,\alpha,\beta)$$

To find C, we need to compute the mode of the likelihood distribution. I made a function to compute the log-likelihood by using the code from the log-likelihood calculation section of the log posterior function. We will use this function with the optimal parameters for α and β we calculated earlier to compute log(C) and then use that result to get C. We find $\log(C) = -260.248$ and $C \approx 9.459 \times 10^{-114}$.

Now that we've defined the target distribution, candidate distribution, C and most importantly, their corresponding log forms, we can define the log-γ function which which is the criterion used to accept/reject a sample value.

$$\gamma(\alpha,\beta) = \frac{\pi(\alpha,\beta)L(x,\alpha,\beta)}{Ch(\alpha,\beta)} = \frac{L(x,\alpha,\beta)}{C}$$

$$\log(\gamma(\alpha,\beta)) = \log(L(x,\alpha,\beta)) - \log(C)$$

We can now sample from the candidate distribution. I sampled 4 million values from U(0,5) and Γ(4,8) each for α and β respectively. We compute the log-γ values using the candidate samples for α and β, and log(C).

We then sample 4 million values from u~U(0,1) and compare log(u) to the log-γ values to see which uniform values fit under our bounds. We keep the candidate samples for α and β at the indices which have met our acceptance criteria i.e. the samples for which $\log(u) < \gamma(\alpha,\beta)$. We are now left with a sample of size of 21459 values from our posterior distribution (an acceptance probability of 5.36×10^{-3}). Obtaining the density of the samples allows us to plot them.
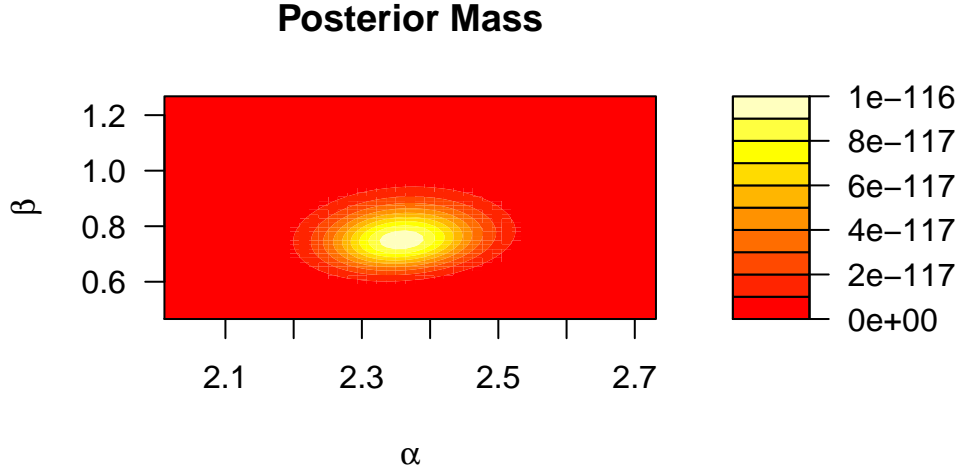
Posterior Mass

Figure 2: *Contour plot of the posterior mass of the accepted samples α and β.*

Figure 2 has its mass in a very similar place to its theoretical counterpart (Figure 1). The mode appears to be at (2.35, 0.75) and the majority of the mass lies between $\alpha\epsilon[2.2, 2.5]$ and $\beta\epsilon[0.6, 95]$. The 3-D posterior distribution plots in Figure 6 (see appendix) also look similar to their theoretical counterparts (see Figure 5, appendix) but it's worth noting the peak in Figure 6 isn't quite as sharp as Figure 5 because the density function uses smaller bounds for α and β because of the distribution of the sample itself.

(b) Estimates Using Accepted Samples

(i) Estimated Location

To estimate the location of the boat, we take the mean of our samples for α and β. The sampled location estimate for the boat is (2.36, 0.768). This isn't largely different from our initial mode of (2.358, 0.75).

(ii) Estimation of $\mathbf{Pr}(\beta > 0.55)$

To estimate $\Pr(\beta > 0.55)$ we divide the number of β samples > 0.55 by the total number of samples. The resulting probability is 0.9989. It is therefore highly likely that $\beta > 0.55$.

(c) Estimating $Pr(2.3 < \alpha < 3.2)$ with Monte Carlo Methods

Understanding the Problem

To generate our sample for α with the probability integral transform, we need to find the inverse of the CDF of our new posterior function given $\beta = 0.5$. This new posterior function is:

$$\pi(\alpha, |\beta = 0.5, x) \propto (\frac{0.5}{A(\alpha, 0.5)})^{200} * \Pi_{i=1}^{200}(\frac{1}{0.5^2 + (x_i - \alpha)^2})$$

I modified the functions for the posterior and log-posterior I used in (a) to take α only as a parameter and remove the terms that became constant given $\beta = 0.5$. The new posterior also has most of the mass concentrated between $\alpha \epsilon [2.1, 2.5]$ (see Figure 7, Appendix).

Using the new posterior function, we need to obtain the inverse of the CDF, $F^{-1}(u)$, which takes in u~U(0,1) and returns a value for α.

Finding an Empirical CDF and its Inverse

The first step to finding the CDF function is to scale the new posterior function such that it becomes a valid PDF (doing this helps prevent underflow later as well). I achieved this by integrating the new posterior over $\alpha \epsilon [0, 5]$. Taking the reciprocal of this gives the proportionality constant, $k = 3.6 \times 10^{117}$, which we can then multiply our new posterior function by to transform it into a PDF. We confirm that it is a PDF by integrating the resulting function over $\alpha \epsilon [0, 5]$ and our result is 1. A plot of the PDF is in the appendix (Figure 8) for interest's sake.

We will use the PDF to find the CDF. Finding the CDF with R's ecdf() function failed because the numbers either side of the large central mass were so small that they were registering as zero. To get around this, I built a CDF() function to calculate an empirical CDF using a million data points along $\alpha \epsilon [0, 5]$. It works by iterating with the following approach: cdf[i]=cdf[i-1] + pdf[alpha[i]]. The final step is to scale the resulting vector by dividing every element by the value of the last index (so final index is = 1). I also created an accompanying function to return $F_\alpha(\alpha)$.
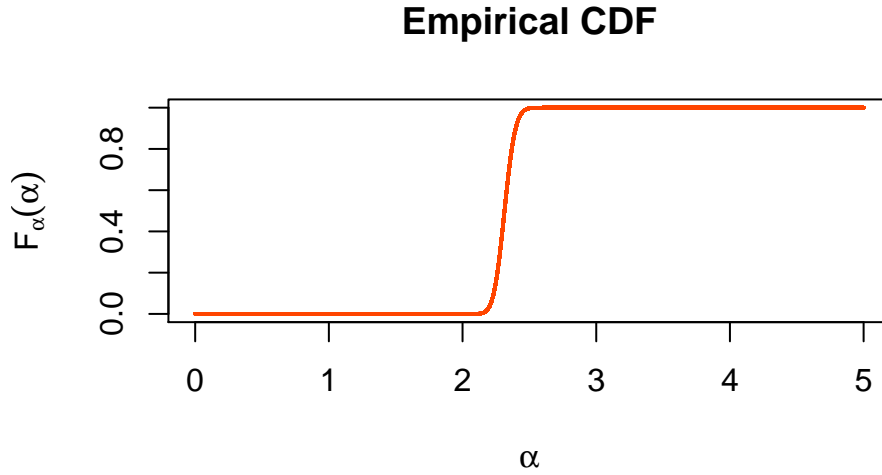
Empirical CDF

Figure 3: Empirical cumulative distribution function for $\pi(\alpha|\beta = 0.5, x)$.

We can confirm the CDF is correct by looking at Figure 3 as it has all the characteristics we'd expect of a CDF for our distribution (S-shape, sharp increase between 2.1 and 2.5, flat elsewhere).

Next, I made a function for the inverse $F^{-1}(u)$. It takes a uniform sample (quartile), u, as a parameter and returns the value of α at that quartile by searching the CDF for the index containing the value closest to the given quartile and computes and returns the corresponding value for α.

Sampling with the Probability Integral Transform Method

We use the inverse function to get our α sample by sampling from U(0,1) and inverse transforming that sample. We will sample 20 000 values for α.
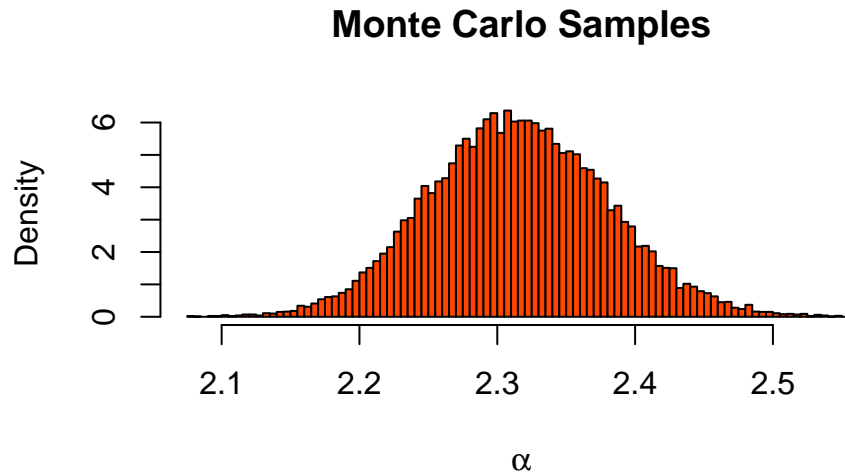
Monte Carlo Samples

Figure 4: Distribution of the samples for α obtained using the probability integral transform method.

Monte Carlo Estimate and Monte Carlo Error

From Figure 4, we can see about half of the mass lies at $\alpha > 2.3$ with no visible mass at $\alpha = 3.2$. The shape of the sample distribution also appears normally distributed about the mean. To get the result using Monte Carlo Methods, we count the number of items in our sample between 2.3 and 3.2 and divide the count by the total samples. The result with our sampled data is $Pr(2.3 < \alpha < 3.2) = 0.5804$. Using R's integrate function with the PDF yields 0.589 and using our empirical CDF $(F_\alpha(3.2) - F_\alpha(2.3))$ yields 0.586 so our Monte Carlo estimate using 20 000 samples isn't too far off. With a larger sample we can expect our accuracy using Monte Carlo sampling to increase.

Using our sample we can compute the Monte Carlo Error using the following formula:

$$SE(\hat{\theta}) = \frac{\sqrt{Var(F^{-1}(u))}}{\sqrt{N}} = \frac{\sqrt{Var(\alpha)}}{\sqrt{N}}$$

The Monte Carlo Error for the sample is 4.57×10^{-4}. With a larger sample we can expect this error to decrease $\propto \frac{1}{\sqrt{N}}$.
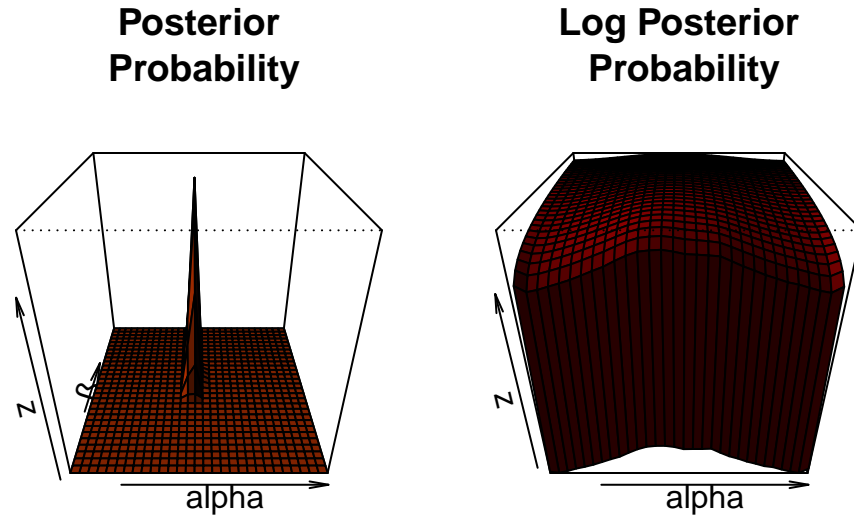
Appendix

Figure 5: *The posterior and log-posterior functions plotted using $\alpha \in$ [1e-8, 5] and $\beta \in$ [1e-8, 1.4]*
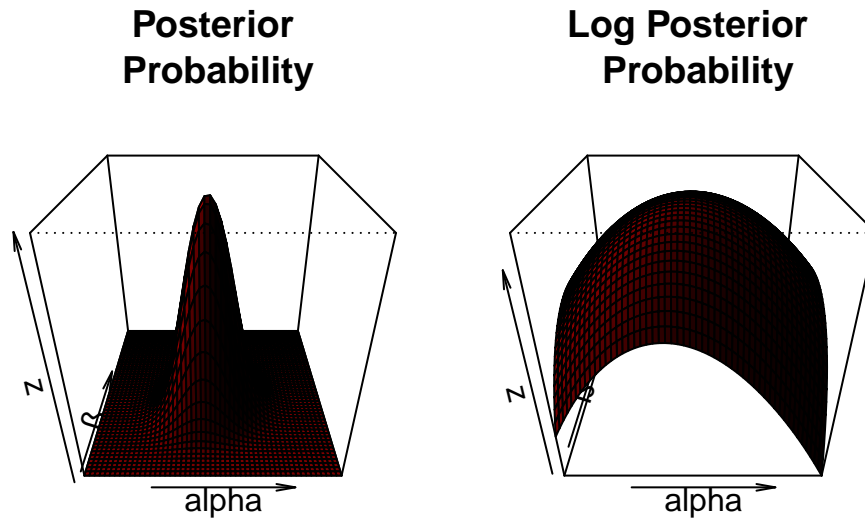
**Posterior
Probability**

**Log Posterior
Probability**

Figure 6: *Density plot of the posterior and log-posterior functions obtained using the accepted values for α and β.*
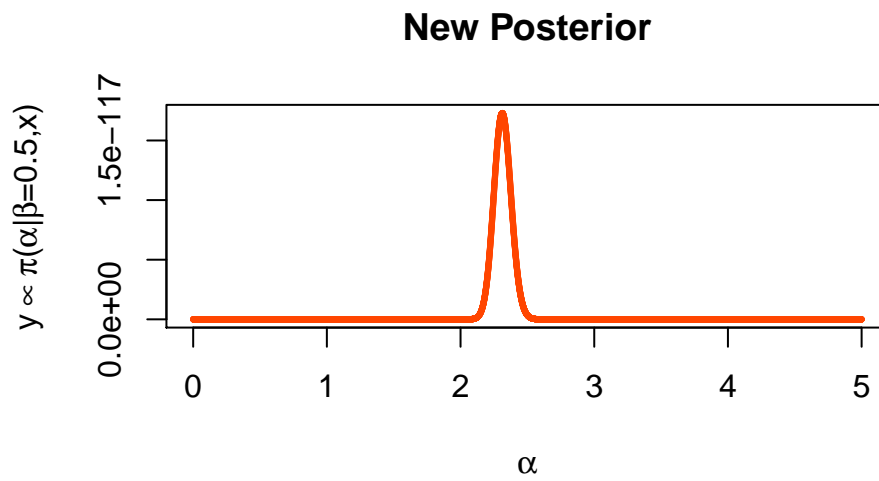
New Posterior

Figure 7: *New posterior distribution, $\pi(\alpha|\beta = 0.5, x)$*
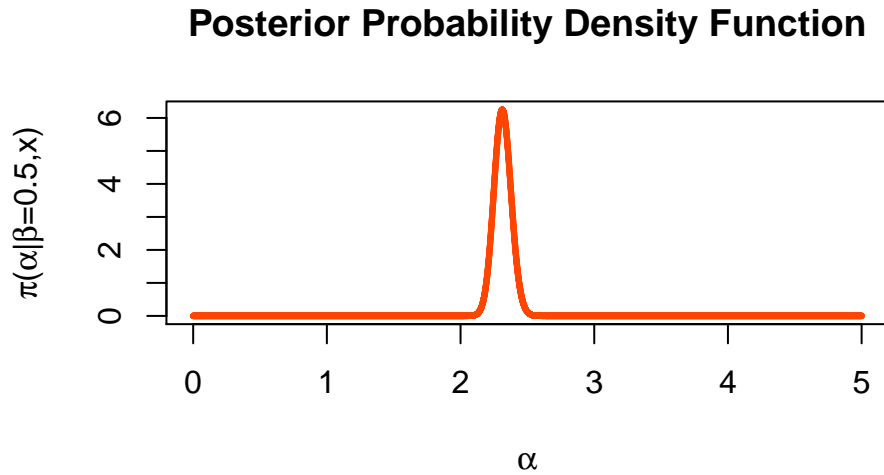
Posterior Probability Density Function

Figure 8: *Scaled probability density function, $\pi(\alpha|\beta = 0.5, x)$*

References

DataMentor. n.d. R 3D Plot (With Examples). [online] Available at: https://www.datamentor.io/r-programming/3d-plot/ [Accessed 10 April 2021].

Mahmoudian, M. and Huber, M., 2017. Fastest way to find nearest value in vector. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/43472234/fastest-way-to-find-nearest-value-in-vector [Accessed 10 April 2021].

Rdocumentation.org. n.d. filled.contour function - Level (Contour) Plots. [online] Available at: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/filled.contour [Accessed 10 April 2021].

Rdocumentation.org. n.d. integrate function - Integration of One-Dimensional Functions. [online] Available at: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/integrate [Accessed 10 April 2021].

Rdocumentation.org. n.d. persp function - Perspective Plots. [online] Available at: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/persp [Accessed 10 April 2021].