Markov Chain Monte Carlo

Matej Miočić[1]

1. Numerical integration vs Monte Carlo integration

Let X be a random variable following the (a,b,c)-PERT distribution with $b > a$, $c > a, a, b, c \in \mathbb{R}$, and the pdf

$$p(x) = \frac{(x-a)^{\alpha-1}(c-x)^{\beta-1}}{B(\alpha,\beta)(c-a)^{\alpha+\beta-1}}$$

supported on $[a,c]$, where

$$\alpha = 1+4\frac{b-a}{c-a}, \quad \beta = 1+4\frac{c-b}{c-a}$$

and $B(x,y)$ is the Beta function. Let $a = 0, b = 10$ and $c = 100$. It turns out that $E[X] = \frac{a+4b+c}{6}$ and $\text{var}[X] = \frac{(E[X]-a)(c-E[X])}{7}$.

1. Use the adaptive trapezoid rule to estimate $E[X]$ to 4 decimal places. How many function evaluations you need to do? (You can use the implementation of the function linked to on the lecture slides. It is written in Matlab.)

   Answer: We need approximately 918 evaluations.

2. Using the Central limit theorem estimate the number of points required to obtain the estimate of $E[X]$ to 2 decimal places using Monte Carlo integration when sampling from the uniform distribution on $[a,c]$.

   Answer: We know that because of Central limit theorem the monte carlo standard error is:

   $$\sigma = \sqrt{\frac{VAR[f(x)]}{n}}. \tag{1}$$

   To estimate the number of points we set the error to 0.01 and calculate the variance of our function and isolate n. By rounding the result we obtain:

   $$n = \left\lfloor \frac{VAR[f(x)]}{\sigma^2} \right\rceil = 2555556. \tag{2}$$

   So the number of estimated points to obtain error of 0.01 is n = 2555556.

3. Verify the estimates above on a few numerical samples.

   Answer:

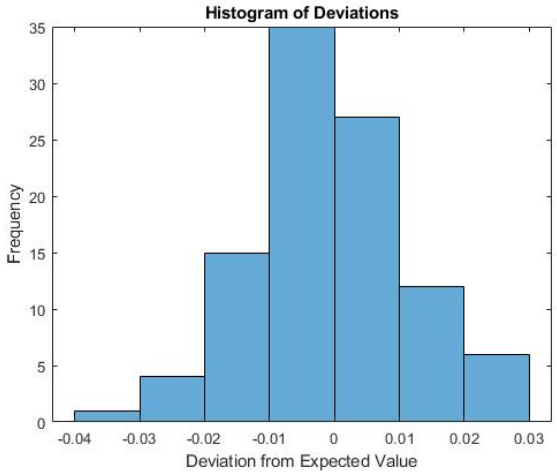

Figure 1. Deviations of 100 runs from the true Expected Value of p.

In Figure 1 we can see that most of the runs of Monte Carlo integration fall in the ± 0.01 deviation from the true Expected Value of p. Standard error of each run is 0.0119.

4. Compare and comment on the results obtained by adaptive trapezoid rule and the Monte Carlo integration.

   Answer: For this example we need a lot more evaluations with the Monte Carlo integration than we do with adaptive trapezoid rule to achieve smaller error. But for higher dimensions trapezoid rule is unfeasible to calculate so Monte Carlo integration is the best thing we have.

2. Importance sampling

We would like to compute the integral

$$I = \int_0^1 x^{-3/4} \cdot e^{-x} dx.$$

1. Plot the integrand $f(x) = x^{-3/4} \cdot e^{-x}$ on $[0,1]$.

   Answer:

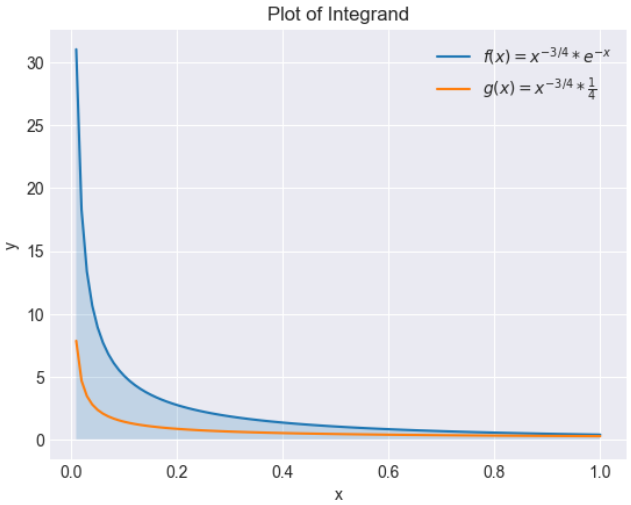

Figure 2. Integrand f(x) (blue curve) and function q(x) (orange curve).

In Figure 2 we show the integrand f(x) and the function q(x) used in 3rd part of this task.

2. Sampling from the uniform distribution estimate I using the Monte Carlo integration with samples of size $n = 10^7$. Compute the average and the standard deviation for 10 samples.

   Answer:

   We ran the estimation 10 times and obtained that the average is: 3.380 ± 0.081.

3. Estimate I as $E_q\left(\frac{f}{q}\right)$ for $q(x) = cx^{-3/4}$ by the following steps:

   (a) Determine c so that q is a density.

   Answer: We know that for a function to be a density its integral on [0,1] needs to evaluate to 1.

$$1 = \int_0^1 q(x)dx = c\int_0^1 x^{-3/4}dx = c4x^{1/4}\Big|_0^1 = 4c,$$
$$c = \frac{1}{4}. \tag{3}$$

   (b) Sample from q using inversion sampling.

   Answer: To sample from q using inversion sampling, first we have to calculate its CDF:

$$F(x) = \int_0^x q(t)dt = \int_0^x \frac{1}{4}t^{-3/4} = t^{1/4}\Big|_0^x = x^{1/4}. \tag{4}$$

   Then we have to calculate its inverse CDF:

$$F^{-1}(u) = u^4. \tag{5}$$

   (c) Repeat step (2) above.

   Answer: By using equation 5 we now generate 10^7 samples and then put them to the power of 4. By running estimation 10 times we obtained that the average is: 3.379 ± 0.000.

4. Compare and comment the results obtained by both methods.

   Answer: By estimating I as $E_q\left(\frac{f}{q}\right)$ for $q(x) = cx^{-3/4}$ we obtained much more accurate results with the same amount of samples. This is because the function q(x) allowed us to sample more often where the density of f(x) is higher, which means that the samples contribute more to the computation of the integral.

3. Markov chains

Let $X_0, X_1, X_2, \ldots$ be an irreducible Markov chain with a finite state space $S = \{w_1, w_2, \ldots, w_m\}$ and a transition matrix K. Note that the irreducibility can be expressed in the following way:

There exist $r \in \mathbb{N}$ and $\varepsilon > 0$ such that for every pair of states $w_i, w_j \in S$

there exists $\ell_{ij} \leq r$ such that $K_{ij}^{\ell_{ij}} > \varepsilon$.

(Recall that A_{ij} stands for the entry in the i-th row and the j-column of A.)

For every state $w_i \in S$ we define the random variable $t_{w_i}^+$, called the hitting time for w_i, by

$$t_{w_i}^+ = \min\{t \geq 1 : X_t = w_i\}.$$

For fixed states $w_i, w_j \in S$ we denote by $P_{w_i}\left(t_{w_j}^+ \in A\right)$ the probability that $t_{w_j}^+ \in A \subseteq \mathbb{N}$ given that $X_0 = w_i$. Similarly, we use $E_{w_i}\left[t_{w_j}^+\right]$ for the expectation of a random variable $t_{w_j}^+$ given that $X_0 = w_i$. Prove the following statements:

1. For $k \in \mathbb{N}$ we have that

$$P_{w_i}\left(t_{w_j}^+ \geq kr\right) \leq (1-\varepsilon)\cdot P_{w_i}\left(t_{w_j}^+ \geq (k-1)r\right) \leq \ldots \leq (1-\varepsilon)^k,$$

   where ε, r are as in (1).

   Answer: We use conditional probability definition:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \tag{6}$$

$$P_{w_i}\left(t_{w_j}^+ \geq kr\right) = P_{w_i}\left(t_{w_j}^+ \geq kr | t_{w_j}^+ \geq (k-1)r\right) * P_{w_i}\left(t_{w_j}^+ \geq (k-1)r\right),$$

   which is less than:

$$(1-\varepsilon) * P_{w_i}\left(t_{w_j}^+ \geq (k-1)r\right),$$

   which is less than:

$$(1-\varepsilon)^2 * P_{w_i}\left(t_{w_j}^+ \geq (k-2)r\right) \ldots$$

   and recursively we come to:

$$(1-\varepsilon)^k$$

4. Metropolis–Hastings algorithm

Let X be a random variable following the (α,η)-Weibull distribution with the parameters $\alpha,\eta \in (0,\infty$, support $x \in (0,\infty)$, and the pdf

$$p(x \mid \alpha,\eta) = \alpha\eta x^{\alpha-1} e^{-x^{\alpha}\eta}$$

Let the prior distribution $\pi(\alpha,\eta)$ be proportional to $e^{-\alpha-2\eta}$. η. We observe the data $x = 0.3, x = 0.5, x = 0.75, x = 0.4$. Approximate the posterior distribution for α and η and compute the mean and the variance of $\alpha - s$ and $\eta - s$ using Metropolis-Hastings algorithm with:

1. a multivariate normal proposal. Use mean $(\alpha,\ \eta)$, while tune the covariance matrix by hand via trial.

2. a proposal distribution

$$q\left(\alpha',\eta' \mid \alpha,\eta\right) = \frac{1}{\alpha\eta} e^{\left(-\frac{\alpha'}{\alpha}-\frac{\eta'}{\eta}\right)}$$

For each of the above scenarios:

1. Generate 5 independent chains of 1000 samples.

   Answer: To generate a chain of 1000 samples we first need to derive the posterior distribution for α and η.

Listing 1. Likelihood

```
likelihood <- function(alpha, eta) {
data <- c(0.3, 0.4, 0.5, 0.75)
ret = 1
for (x in data) {
    ret <- (alpha * eta * x^(alpha - 1) *
         exp(-x^alpha * eta) ) * ret
}
return(ret)
}
```

Listing 2. Prior

```
prior <- function(alpha, eta) {
    exp(-alpha - 2 * eta) * eta
}
```

Listing 3. Posterior

```
posterior <- function(x) {
    likelihood(x[1], x[2]) * prior_pdf(x
        [1], x[2])
}
```

Then we need to define proposal functions:

Listing 4. Multivariate normal proposal

```
norm_proposal <- function(x, sigma) {
    rmvnorm(1, x, sigma = sigma)
}
```

Listing 5. Proposal q

```
q_proposal <- function(alpha, eta) {
    alpha_ <- rexp(1, 1 / alpha)
    eta_ <- rexp(1, 1 / eta)
    return(c(alpha_, eta_))
}
```

With the proposals and posterior defined, we can now run Metropolis–Hastings algorithm. We show results in the following subtasks.

2. Apply standard MCMC diagnostics for each algorithm/run (traceplot for each parameter and all chains at the same time), autocovariance and the ESS.

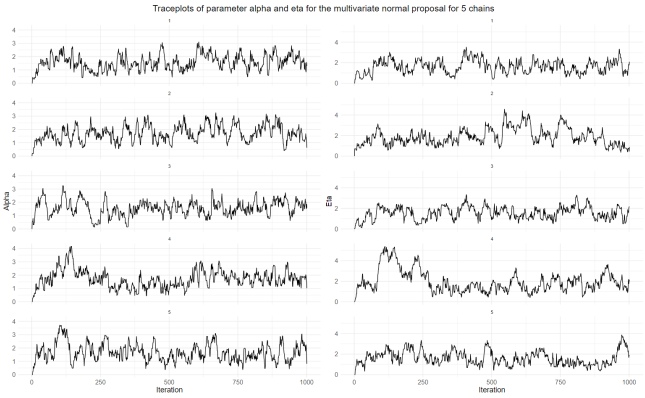

Figure 3. Traceplots of parameter alpha and eta for the **multivariate normal proposal** for 5 chains.

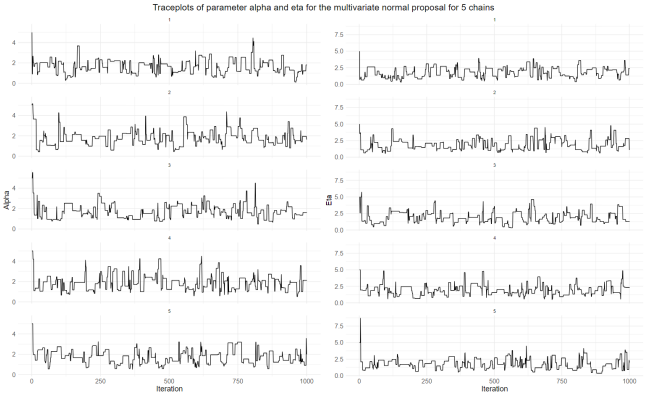

Figure 4. Traceplots of parameter alpha and eta for the **second proposal q(x)** for 5 chains.

Answer: In Figures 3 and 4 we plot traceplots for both alpha and eta parameters for multivariate and q(x) proposals respectfully for each chain. We can see that the traceplots for both parameters using q(x) proposals look better than for multivariate normal; the chains have smoother, less erratic lines.

In Figures 5 and 6 we show autocorrelation plots for both alpha and eta parameters for multivariate and q(x) proposals respectfully for each chain. We can see that

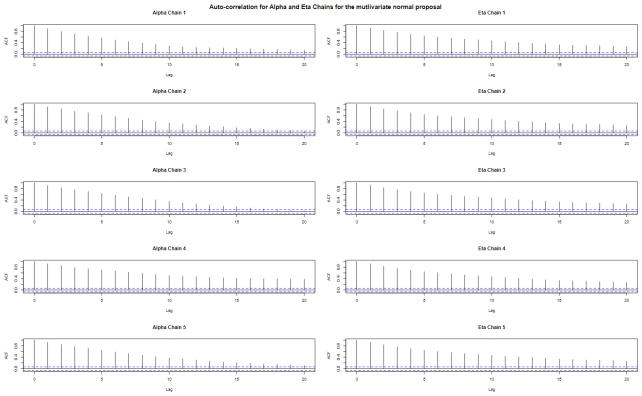

Figure 5. Auto-correlation for Alpha and Eta Chains for the **multivariate normal proposal**.

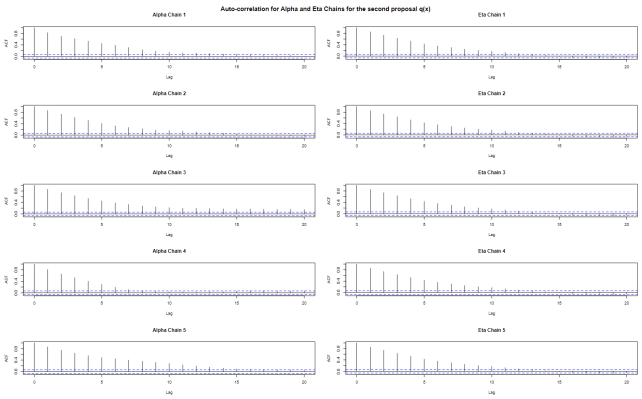

Figure 6. Auto-correlation for Alpha and Eta Chains for the **second proposal q(x)**.

the autocorrelation for chains for both parameters is lower when using q(x) as the proposal.

In Figures 7 and 8 we show effective sample size for both proposals. We can see that for both parameters and all chains the effective sample size is higher for q(x) proposal than for multivariate normal.

In Figures 9 and 10 we show the mean and their standard error for chain samples for both proposals. We can see that when we use proposal q(x) the error is smaller and the means for each chain are more similar to each other compared to samples using multivariate normal proposal.

3. Compare both algorithms in sampling from the target distribution. The bivariate normal distribution is not a good proposal to sample from this distribution. Comment why this holds. The proposal q is a good proposal. Comment why this holds. Is the sample size 1000 large enough?

   Answer: To tune the parameters of MCMC easier we plotted out target distribution shown in Figure 11.

   Now that we have an idea how the shape of of posterior distribution looks like, we can tune the parameters of

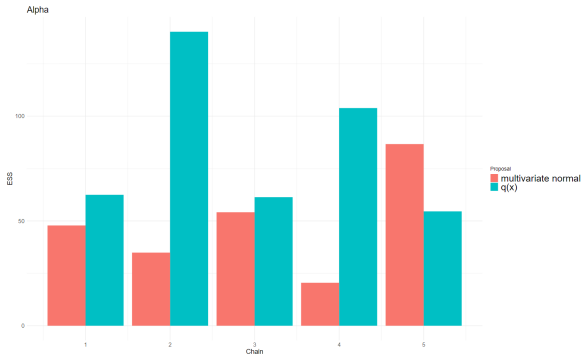

Figure 7. Effective sample size for **Alpha** Chains for both proposals.

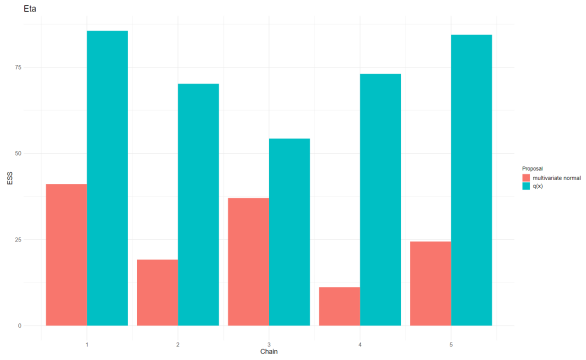

Figure 8. Effective sample size for **Eta** Chains for both proposals.

multivariate normal and q(x) easier to match the shape. We decided to go with multivariate normal with mean in (α, η) and covariance matrix:

$$\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

Contour plot of this multivariate normal is shown in Figure 12.

And for the second proposal we went with initial parameters $\alpha = 3$ and $\eta = 3$. This distribution is shown in Figure 13.

We can see that the multivariate distribution is somewhat similar to out posterior distribution, which gives us the impression that it works good. On the contrary we often sample out of our posterior distribution bounds (0, inf), which we can fix by rejecting samples out of bounds but this makes us reject a lot of samples. We also often "wander" away from the distribution towards bigger values, which don't contribute a lot. On the other hand the second proposal - q(x) samples more where samples have lower value and does not go out of range. This makes it a better proposal than multivariate normal.

We conclude that 1000 samples with the multivariate normal proposal is not enough to approximate this posterior function. But for the second proposal our error of

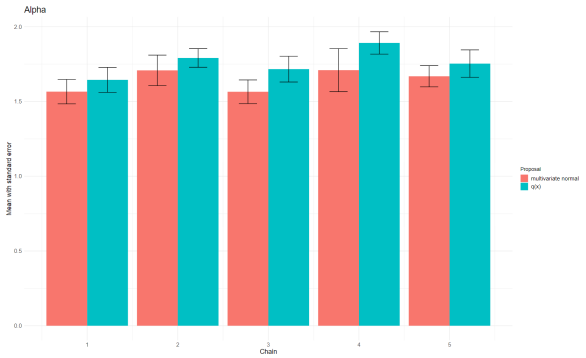

Figure 9. Mean of **Alpha** with its standard error for each chain and both proposals.

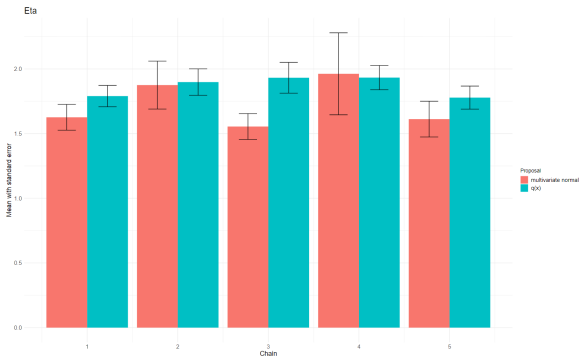

Figure 10. Mean of **Eta** with its standard error for each chain and both proposals.

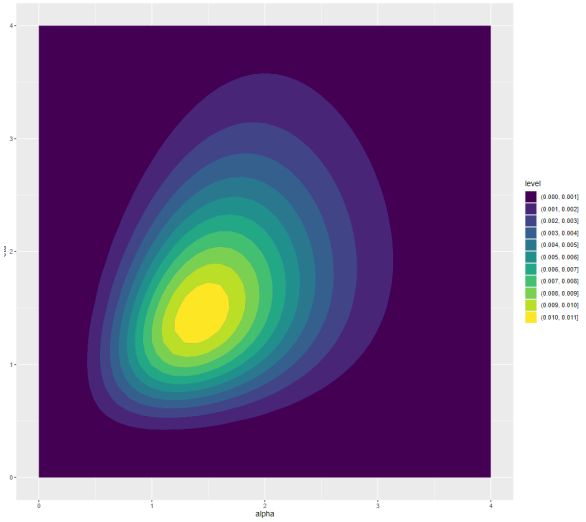

Figure 11. Contour plot of our target - posterior distribution $f(\alpha,\eta)$.

the samples is pretty small so we could say that 1000 samples suffices.

4. Estimate the probability $(\alpha,\eta) \in [1.3,\infty) \times [1.3,\infty)$.

   Answer: To obtain the probability we just count how many samples are above 1.3 and divide by the number of all samples for each chain.

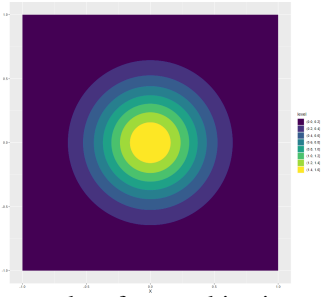

Figure 12. Contour plot of our multivariate normal proposal with μ = [0,0] and $\Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$.

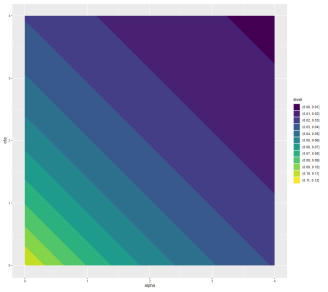

Figure 13. Contour plot of our second proposal q(x) with α = 3 and η = 3.

(a) When using **multivariate normal** for our proposal we get:

$$P(\alpha > 1.3, \eta > 1.3) = 0.52\% \pm 0.07. \quad (7)$$

(b) When using **q(x)** for our proposal we get:

$$P(\alpha > 1.3, \eta > 1.3) = 0.55\% \pm 0.06. \quad (8)$$

From the results we see that we got similar percentages of α and η. With more samples we could decrease our standard deviation and get more accurate results.