L9: Dependent MC Sampling, Metropolis-Hastings & MCMC Diagnostics

1 Metropolis-Hastings Algorithm: Example with Hurricane Event Time Data

To demonstrate the Metropolis-Hastings algorithm in a simple case, we return to the time between hurricane data in Lecture 6 where the Weibull distribution was the likelihood for shape parameter α, and a Gamma distribution was used as the prior for α. Then the posterior:

$$p(\alpha|\boldsymbol{y}) \propto \pi(\alpha) f(\boldsymbol{y}|\alpha) = \frac{\lambda^\kappa}{\Gamma(\kappa)} \alpha^{\kappa-1} \exp(-\lambda\alpha) \times \alpha^n \exp\left(-\sum_{i=1}^n y_i^\alpha\right) \prod_{i=1}^n y_i^{\alpha-1} \tag{1}$$

With the Metropolis-Hastings algorithm, the normalising constant can be ignored. In addition all terms in the joint distribution not involving α can be ignored.

Pseudo-code for the t^{th} iteration of the algorithm is the following:

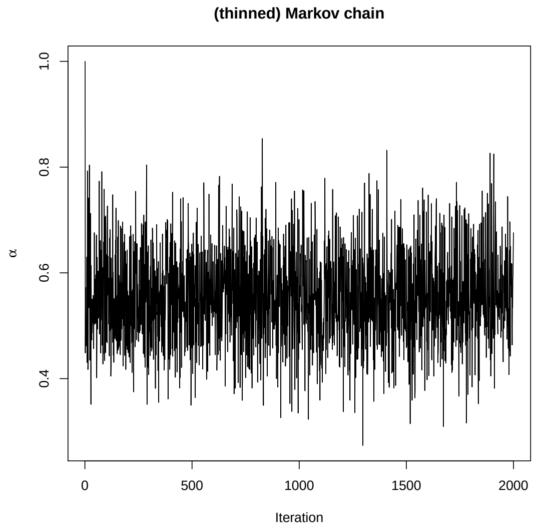
1. Generate candidate value, α^c, from the proposal, $q(\alpha|\alpha^{t-1})$.
2. Evaluate the MHR:

$$MHR(\alpha^{t-1}, \alpha^c) = \frac{(\alpha^c)^{\kappa-1} \exp(-\lambda\alpha^c) \times (\alpha^c)^n \exp\left(-\sum_{i=1}^n y_i^{\alpha^c}\right) \prod_{i=1}^n y_i^{\alpha^c-1}}{(\alpha^{t-1})^{\kappa-1} \exp(-\lambda\alpha^{t-1}) \times (\alpha^{t-1})^n \exp\left(-\sum_{i=1}^n y_i^{\alpha^{t-1}}\right) \prod_{i=1}^n y_i^{\alpha^{t-1}-1}} \times \frac{q(\alpha^{t-1}|\alpha^c)}{q(\alpha^c|\alpha^{t-1})}$$

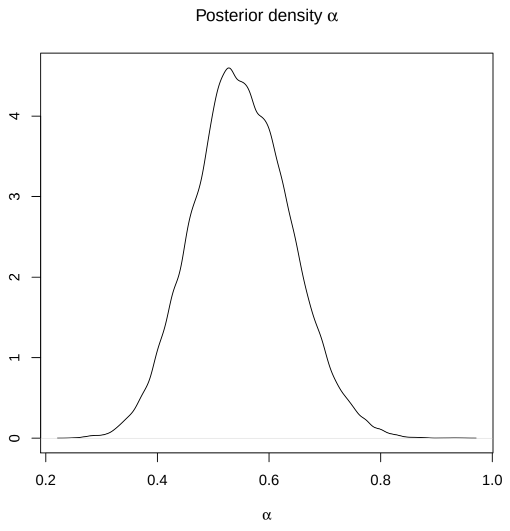
3. Generate a Uniform(0,1) random variable, u^*.
   If $u^* \leq \min(1, MHR)$, set α^t=α^c, else set α^t=α^{t-1}.

R code to generate a sample from the posterior using a $q(\alpha|\alpha^{t-1})$ = Gamma(2,3) proposal distribution is in Appendix A.1. A portion of the code is shown below. The MHR has been calculated on the log scale first to lessen the chance of under- or over-flows.

```
tally <- tally+1
candidate <- rgamma(n=1,shape=a.q,rate=b.q)
previous  <- alpha.star[tally-1]

log.joint.cand <- (n+kappa-1)*log(candidate) -  sum(hurricane.gaps^candidate) -
                  lambda*candidate +   (candidate-1)*sum(log(hurricane.gaps))
log.joint.prev <- (n+kappa-1)*log(previous) -  sum(hurricane.gaps^previous) -
                  lambda*previous +   (previous-1)*sum(log(hurricane.gaps))
log.q.cand     <- dgamma(candidate,a.q,b.q,log=TRUE)
log.q.prev     <- dgamma(previous,a.q,b.q,log=TRUE)
MHR            <- exp(log.joint.cand + log.q.prev - log.joint.prev - log.q.cand)

u.star         <- runif(1)
if(u.star <= min(1,MHR)) {
  # keep the candidate
  alpha.star[tally] <- candidate
  number.accept <- number.accept+1
} else {
  # reject the candidate
  alpha.star[tally] <- previous
}
```



(a) Trace Plot of α for the hurricane event time data based on a Weibull(α,β = 1) distribution with a Gamma(0.1,0.1) prior and Gamma(2,3) proposal distribution. Every 50th value is shown.



(b) Posterior distribution for α for the hurricane event time data based on a Weibull(α,β = 1) distribution with a Gamma(0.1,0.1) prior and Gamma(2,3) proposal distribution. B=1000.

A Markov chain of length N=100,000 was generated with an acceptance rate of 0.26. The resulting α values (thinned to show every 50th value) are plotted in Figure 1a and the posterior distribution is plotted in Figure 1b.

Using a Burn-in B=1000, the following summary statistics were calculated:

```
summary(alpha.star)
#     Min. 1st Qu.  Median    Mean   3rd Qu.    Max.
#  0.2443  0.4919   0.5489   0.5512  0.6097   0.9477
```

Arbitrary probabilities can be calculated as well, such as $\Pr(0.40 \leq \alpha \leq 0.71)$:

$$\widehat{\Pr}(0.40 \leq \alpha \leq 0.71) = \frac{1}{N-B} \sum_{t=B+1}^{N} I(0.40 \leq \alpha^t \leq 0.71)$$

Using R:

```
 sum(alpha.star>=0.40 & alpha.star<=0.71)/(N-Burnin) # 0.9257273
```

2 Metropolis-Hastings Algorithm: Why it works

We will apply the results from Lecture Notes 8 on discrete time, discrete state space Markov chains to include discrete time, continuous state space Markov chains, but we do not explain or justify the theory that makes this acceptable, however.

We will consider the case where the generated random variable, θ, is a scalar.

Recall from Lecture Notes 8 that:

1. Reversible Markov chains, namely,

$$\pi(i)\Pr(\theta^t = j|\theta^{t-1} = i) = \pi(j)\Pr(\theta^t = i|\theta^{t-1} = j) \tag{2}$$

   have stationary distributions.

2. If an irreducible, aperiodic Markov chain has a stationary distribution, π, then that stationary distribution is unique and is the limiting distribution:

$$\lim_{n\to\infty} \Pr(\theta^{t+n} = j|\theta^t = i) = \pi(j) \tag{3}$$

Thus if a Markov chain is irreducible, aperiodic, and reversible, then it has a unique stationary and limiting distribution.

Claim. The Metropolis-Hastings Algorithm satisfies the above two conditions and is constructed such that the stationary and limiting distribution is the target distribution.

Proof. (This is based on *Introduction to Probability Models, 8th Ed.*, S. Ross.) This proof assumes a discrete, countable state space. Without loss of generality, let the state space be the set of positive integers, $S=\{1, 2, \ldots, \}$. Let $p(\theta)$ be a probability mass function with $\theta \in S$.

- Let θ^t be a stochastic process (a time indexed random variable) and,
- Let Q be any specified irreducible Markov tpm on S where $q(j|i)$ is the jth column of row i.
- Given $\theta^{t-1}=o$, where $o \in S$ (and stands for "old"), generate a random variable Y such that $Y=c$ with probability $q(c|o)$, where $c \in S$ (and stands for "candidate").
- If $Y=c$, set $\theta^t = c$ with probability $\alpha(o, c)$, otherwise set $\theta^t = o$. The process θ^t is thus a Markov chain:

$$\begin{aligned}\Pr(\theta^t = c|\theta^{t-1} = o) &= q(c|o)\alpha(o, c), c \neq o \\ \Pr(\theta^t = o|\theta^{t-1} = o) &= q(o|o) + \sum_{k\neq o} q(k|o)(1 - \alpha(o, k))\end{aligned}$$

- θ^t will be a reversible Markov chain with stationary probability mass function $p(\theta)$ if

$$p(o)\Pr(\theta^t = c|\theta^{t-1} = o) = p(c)\Pr(\theta^t = o|\theta^{t-1} = c), \quad \forall c \neq o \tag{4}$$

- Define $\alpha(o, c)$ as follows[1]:

$$\alpha(o, c) = \min\left(1, \frac{p(c)q(o|c)}{p(o)q(c|o)}\right) \tag{5}$$

[1]This was the ingenious part.

- There are two cases to consider:
  - Case 1: $p(c)q(o|c) < p(o)q(c|o)$. Then

    $$\alpha(o,c) = \frac{p(c)q(o|c)}{p(o)q(c|o)} \quad \text{and} \quad \alpha(c,o) = 1$$

    and

    $$\begin{aligned} p(o)\Pr(\theta^t = c|\theta^{t-1} = o) &= p(o)q(c|o)\alpha(o,c) = p(o)q(c|o)\frac{p(c)q(o|c)}{p(o)q(c|o)} \\ &= p(c)q(o|c) = p(c)q(o|c)\alpha(c,o) = p(c)\Pr(\theta^t = o|\theta^{t-1} = c) \end{aligned}$$

  - Case 2: $p(c)q(o|c) \geq p(o)q(c|o)$. Then

    $$\alpha(o,c) = 1 \quad \text{and} \quad \alpha(c,o) = \frac{p(o)q(c|o)}{p(c)q(o|c)}$$

    and

    $$\begin{aligned} p(c)\Pr(\theta^t = o|\theta^{t-1} = c) &= p(c)q(o|c)\alpha(c,o) = p(c)q(o|c)\frac{p(o)q(c|o)}{p(c)q(o|c)} \\ &= p(o)q(c|o) = p(o)q(c|o)\alpha(o,c) = p(o)\Pr(\theta^t = c|\theta^{t-1} = o) \end{aligned}$$

  Thus θ^t is a reversible Markov chain with stationary probability $p(\theta)$.

3 Metropolis-Hastings Algorithm: Special case proposals

3.1 Random walk proposals

A random walk proposal has the following structure.

$$\theta^c = \theta^o + \epsilon$$

where θ^o is the old value and θ^c is the candidate value and ϵ is a mean zero random variable with probability distribution, f. Some special cases:

$$\begin{aligned}
&\epsilon \sim \text{Normal}(0, \sigma^2) \\
&\epsilon \sim t_{2\ df} \\
&\epsilon \sim \text{Uniform}(-b, b)
\end{aligned}$$

These three cases are examples of *symmetric* proposal distributions:

$$q(\theta^c|\theta^o) = q(\theta^o|\theta^c) \tag{6}$$

For example, if $\epsilon \sim$ Normal(0,σ^2),

$$q(\theta^c|\theta^o) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta^c - \theta^o)^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta^o - \theta^c)^2\right) = q(\theta^o|\theta^c)$$

When the proposal is symmetric,

$$\frac{q(\theta^o|\theta^c)}{q(\theta^c|\theta^o)} = 1$$

and the Metropolis Hastings Ratio simplifies to

$$MHR(\theta^o, \theta^c) = \frac{p(\theta^c)q(\theta^o|\theta^c)}{p(\theta^o)q(\theta^c|\theta^o)} = \frac{p(\theta^c)}{p(\theta^o)} \tag{7}$$

Note symmetric distributions often readily extend to multivariate settings, e.g, use a Bivariate normal as a proposal for Θ=(θ_1, θ_2).

Metropolis sampler. Random walk proposals are a special case of a Metropolis sampler. A Metropolis sampler is one where $q(\theta^c|\theta^o) = q(\theta^o|\theta^c)$ and thus $q(\theta^o|\theta^c)/q(\theta^c|\theta^o) = 1$, and thus the proposal need not be evaluated. An example of a Metropolis sampler that is *not* a random walk is $\theta^c = \theta^o + \epsilon$, $\epsilon \sim$ Uniform(-1, 2), and the pdf for $\theta^c|\theta^o$ = pdf for $\theta^o|\theta^c$ = 1/3, so long at the support for θ is unbounded.

Transforming parameters to facilitate random walk proposals. When the support of a parameter θ is restricted, e.g., $\theta > 0$, then a random walk proposal will be wasteful when a proposed value is outside of the support of θ. For example, suppose $\theta > 0$, and the proposal is Uniform($\theta^o - 1$, $\theta^o + 1$). Suppose θ^o=0.2, then the proposal is Uniform(-0.8,1.2) and there is a 0.8/2 = 0.4 probability of getting $\theta^c \leq 0$. If $\theta^c < 0$, $p(\theta^c)$=0, so the value will be rejected.

A less wasteful procedure is to transform a range restricted parameter with an invertible mapping to the real number line, say ϕ=$g(\theta)$, where g is a 1:1 function, and then applying a random walk proposal to the transformed parameter. The prior distribution for the transformed parameter is then

$$\pi_\phi(\phi) = \pi_\theta(g^{-1}(\phi)) \left|\frac{dg^{-1}}{d\phi}\right|$$

For example, the sampling distribution for the data, y, is Exponential(θ) distribution, θ must be greater than 0. Suppose the prior distribution for θ is Gamma(α, β). Define $\phi = g(\theta) = \ln(\theta)$, then $g^{-1}(\phi) = \exp(\phi)$ and $dg^{-1}(\phi)/d\phi = \exp(\phi)$. The prior distribution for ϕ:

$$\begin{aligned}\pi_\phi(\phi) &= \frac{\beta^\alpha}{\Gamma(\alpha)}(\exp(\phi))^{\alpha-1}\exp(-\beta * \exp(\phi)) \times \exp(\phi) \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)}(\exp(\phi))^{\alpha}\exp(-\beta * \exp(\phi))\end{aligned}$$

The data distribution (the likelihood) will need to be rewritten in terms of ϕ, i.e., substitute $g^{-1}(\phi)$ for θ. Then use random walk proposal for ϕ such as

$$\phi^c = \phi^o + \epsilon_t$$

where $\epsilon_t \sim$ Normal(0,0.5), and one is left with a Metropolis sampler where the proposal distributions are not evaluated for the MHR.

3.2 Independence proposals

An independence proposal is one where the proposal does not depend on the previous value, θ^o. For example, $\theta^c \sim$ Normal(0,10). The example of the Weibull given in Section 1 of the Gamma(2,3) proposal was an independence proposal.

The MHR is then:

$$MHR(\theta^o, \theta^c) = \frac{p(\theta^c)q(\theta^o)}{p(\theta^o)q(\theta^c)}$$

Note that the generated values are still coming from a Markov chain because the transition density or kernel is the product of the proposal density and the Metropolis-Hastings ratio.

4 Multidimensional Θ: One-at-a-time or Single updates

In the case of multidimensional Θ=($\theta_1, \theta_2, \ldots, \theta_k$), the entire vector can be updated at once, or subsets ("blocks") can be updated, or each individual parameter, θ_i, i=1,2,...,q can be updated one-at-a time. The one-at-a-time update leads to some simplification in the calculation of the Metropolis Hasting ratio.

For example, let Θ= ($\theta_1, \theta_2, \theta_3$). Set the initial values, ($\theta_1^0, \theta_2^0, \theta_3^0$). The values will be updated in the order 1, 2, and 3. First generate a candidate value for θ_1 from $q(\theta_1^c|\theta_1^0, \theta_2^0, \theta_3^0)$, then evaluate the MHR:

$$\begin{aligned}MHR(\theta_1^0, \theta_1^c) &= \frac{p(\theta_1^c, \theta_2^0, \theta_3^0)q(\theta_1^0|\theta_1^c, \theta_2^0, \theta_3^0)}{p(\theta_1^0, \theta_2^0, \theta_3^0)q(\theta_1^c|\theta_1^0, \theta_2^0, \theta_3^0)} \\ &= \frac{p(\theta_1^c|\theta_2^0, \theta_3^0)p(\theta_2^0, \theta_3^0)q(\theta_1^0|\theta_1^c, \theta_2^0, \theta_3^0)}{p(\theta_1^0|\theta_2^0, \theta_3^0)p(\theta_2^0, \theta_3^0)q(\theta_1^c|\theta_1^0, \theta_2^0, \theta_3^0)} \\ &= \frac{p(\theta_1^c|\theta_2^0, \theta_3^0)q(\theta_1^0|\theta_1^c, \theta_2^0, \theta_3^0)}{p(\theta_1^0|\theta_2^0, \theta_3^0)q(\theta_1^c|\theta_1^0, \theta_2^0, \theta_3^0)}\end{aligned}$$

Note the cancellation of the joint distribution of θ_2^0, θ_3^0. Thus only terms involving θ_1 in $p(\theta_1, \theta_2, \theta_3)$ need to evaluated when calculating the MHR. Generate U^* from Uniform(0,1) and set $\theta_1^1 = \theta_1^c$ if $U^* \leq \min(1, MHR(\theta_1^0, \theta_1^c))$, else set θ_1^1=θ_1^0.

Next generate a candidate value for θ_2 from $q(\theta_2^c|\theta_1^1, \theta_2^0, \theta_3^0)$. Then evaluate the MHR:

$$\begin{aligned} MHR(\theta_2^0, \theta_2^c) &= \frac{p(\theta_1^1, \theta_2^c, \theta_3^0) q(\theta_2^0|\theta_1^1, \theta_2^c, \theta_3^0)}{p(\theta_1^1, \theta_2^0, \theta_3^0) q(\theta_2^c|\theta_1^1, \theta_2^0, \theta_3^0)} \\ &= \frac{p(\theta_2^c|\theta_1^1, \theta_3^0) q(\theta_2^0|\theta_1^1, \theta_2^c, \theta_3^0)}{p(\theta_2^0|\theta_1^1, \theta_3^0) q(\theta_2^c|\theta_1^1, \theta_2^0, \theta_3^0)} \end{aligned}$$

Again generate U^* from Uniform(0,1) and set $\theta_2^1 = \theta_2^c$ if $U^* \leq \min(1, MHR(\theta_2^0, \theta_2^c))$, else set $\theta_2^1 = \theta_2^0$.

To complete the update at iteration 1: generate a candidate value for θ_3 from $q(\theta_3^c|\theta_1^1, \theta_2^1, \theta_3^0)$. Then evaluate the MHR:

$$\begin{aligned} MHR(\theta_3^0, \theta_3^c) &= \frac{p(\theta_1^1, \theta_2^1, \theta_3^c) q(\theta_3^0|\theta_1^1, \theta_2^1, \theta_3^c)}{p(\theta_1^1, \theta_2^1, \theta_3^0) q(\theta_3^c|\theta_1^1, \theta_2^1, \theta_3^0)} \\ &= \frac{p(\theta_3^c|\theta_1^1, \theta_2^1) q(\theta_3^0|\theta_1^1, \theta_2^1, \theta_3^c)}{p(\theta_3^0|\theta_1^1, \theta_2^1) q(\theta_3^c|\theta_1^1, \theta_2^1, \theta_3^0)} \end{aligned}$$

Again generate U^* from Uniform(0,1) and set $\theta_3^1 = \theta_3^c$ if $U^* \leq \min(1, MHR(\theta_3^0, \theta_3^c))$, else set $\theta_3^1 = \theta_3^0$.

5 Metropolis-Hastings example with 2 parameters

To demonstrate Metropolis-Hastings in the multiparameter case, we will use the between event times hurricane data but now estimate both parameters of the Weibull distribution, α and β. The Weibull distribution pdf is:

$$f(y|\alpha, \beta) = \frac{\alpha}{\beta} \left(\frac{y}{\beta} \right)^{\alpha - 1} \exp\left(-(y/\beta)^{\alpha} \right) \tag{8}$$

The joint distribution for a sample of size n:

$$f(\boldsymbol{y}|\alpha, \beta) = \left(\frac{\alpha}{\beta} \right)^n \prod_{i=1}^{n} \left(\frac{y_i}{\beta} \right)^{\alpha - 1} \exp\left[-\sum_{i=1}^{n} (y_i/\beta)^{\alpha} \right] \tag{9}$$

For simplicity, we will use identical, and independent Gamma(κ,λ) priors for α and β. Thus the prior pdf:

$$\pi(\alpha, \beta) = \left(\frac{\lambda^{\kappa}}{\Gamma(\kappa)} \right)^2 \alpha^{\kappa-1} \beta^{\kappa-1} \exp(-\lambda\alpha) \exp(-\lambda\beta) \propto (\alpha\beta)^{\kappa-1} \exp(-\lambda(\alpha + \beta)) \tag{10}$$

The normalising constants can be ignored because they will cancel in the MHR.

To lessen the chance of underflows/overflows we will work with log transformed values in the Metropolis Hasting ratio and then exponentiate at the end. The log of the product of the priors and the likelihood is then:

$$\ln\left[\pi(\alpha, \beta) f(\boldsymbol{y}|\alpha, \beta)\right] = (\kappa - 1)\ln(\alpha\beta) - \lambda(\alpha + \beta) + n \ln(\alpha/\beta) + \sum_{i=1}^{n} (\alpha - 1) \ln(y_i/\beta) - \sum_{i=1}^{n} (y_i/\beta)^{\alpha} \tag{11}$$

For the proposals we will use a normal random walk and update both parameters simultaneously (as a pair):

$$\alpha^c \sim \text{Normal}\left(\alpha^{t-1}, \sigma_{\alpha}\right) \tag{12}$$

$$\beta^c \sim \text{Normal}\left(\beta^{t-1}, \sigma_{\beta}\right) \tag{13}$$

If either of the candidate values are less than or equal to zero, they will be rejected. The values of σ_α and σ_β will be experimented with to keep the probability of rejection relatively low.

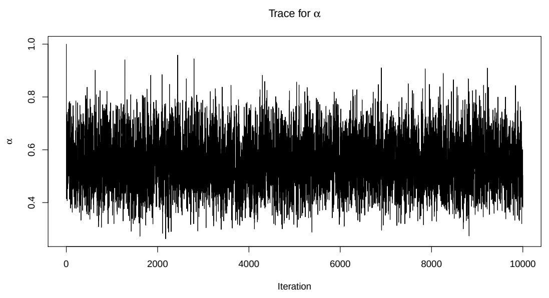
With a symmetric proposal, we have a Metropolis algorithm and the MHR simplifies to the following:

$$MHR([\alpha^{t-1},\beta^{t-1}],[\alpha^c,\beta^c]) = \exp\Bigg[(\kappa-1)\ln(\alpha^c\beta^c) - \lambda(\alpha^c+\beta^c) + n\ln(\alpha^c/\beta^c) + \sum_{i=1}^{n}(\alpha^c-1)\ln(y_i/\beta^c) - \sum_{i=1}^{n}(y_i/\beta^c)^{\alpha^c} \\ - (\kappa-1)\ln(\alpha^{t-1}\beta^{t-1}) + \lambda(\alpha^{t-1}+\beta^{t-1}) - n\ln(\alpha^{t-1}/\beta^{t-1}) - \sum_{i=1}^{n}(\alpha^{t-1}-1)\ln(y_i/\beta^{t-1}) + \sum_{i=1}^{n}(y_i/\beta^{t-1})^{\alpha^{t-1}}\Bigg] \tag{14}$$

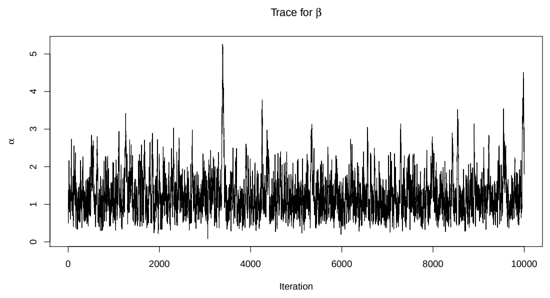
Complete R code for the sampler is shown in Appendix A.2. A portion of the code is shown below.

```
tally         <- 1
number.accept <- 0
while(tally < N) {
  tally <- tally+1
  # Proposals are Normal(alpha^{-1},sigma.alpha), Normal(beta^{-1},sigma.beta), thus
  # symmetric (random walk)
  alpha.c  <- rnorm(n=1,mean=alpha.star[tally-1],sd=sigma.alpha)
  beta.c   <- rnorm(n=1,mean=beta.star[tally-1],sd=sigma.beta)

  if(alpha.c < 0 || beta.c < 0) {
     alpha.star[tally] <- alpha.star[tally-1]
     beta.star[tally]  <- beta.star[tally-1]

  } else {

  alpha.p <- alpha.star[tally-1]
  beta.p  <- beta.star[tally-1]

  log.joint.cand <- (kappa-1)*log(alpha.c*beta.c)-lambda*(alpha.c+beta.c) +
     n*log(alpha.c/beta.c) + (alpha.c-1)*sum(log(y/beta.c)) -
     sum((y/beta.c)^alpha.c)

  log.joint.prev <- (kappa-1)*log(alpha.p*beta.p)-lambda*(alpha.p+beta.p) +
  n*log(alpha.p/beta.p) + (alpha.p-1)*sum(log(y/beta.p)) -
  sum((y/beta.p)^alpha.p)

  MHR            <- exp(log.joint.cand - log.joint.prev)
  u.star         <- runif(1)
  if(u.star <= min(1,MHR)) {
    # keep the candidate
    alpha.star[tally] <- alpha.c
    beta.star[tally]  <- beta.c
    number.accept <- number.accept+1
  } else {
    # reject the candidate
    alpha.star[tally] <- alpha.p
    beta.star[tally]  <- beta.p
   }
  }
}
```

The trace plots for α and β are shown in Figures 2a and 2b.

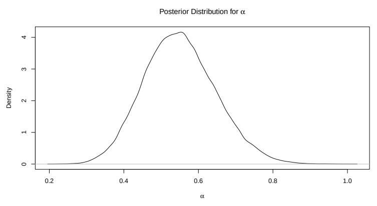


(a) Markov chain of α for the hurricane event time data based on a Weibull(α,β) distribution with a Gamma(0.1,0.1) prior and normal random walk proposal.

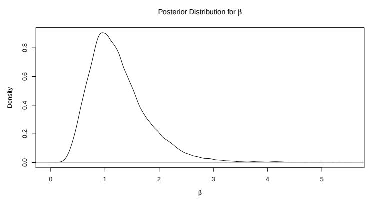


(b) Markov chain of β for the hurricane event time data based on a Weibull(α,β) distribution with a Gamma(0.1,0.1) priors and normal random walk proposal.

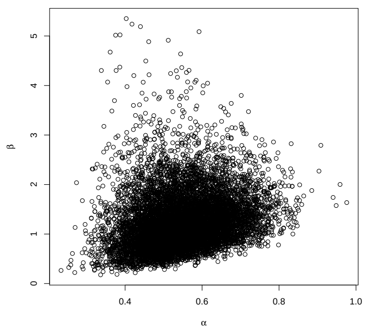
The posterior densities (without burn-in) are shown in Figures 3a and 3b.



(a) Posterior distribution for α for the hurricane event time data based on a Weibull(α,β) distribution with a Gamma(0.1,0.1) prior and and normal random walk proposal.



(b) Posterior distribution for β for the hurricane event time data based on a Weibull(α,β) distribution with a Gamma(0.1,0.1) prior and and normal random walk proposal.



(c) Scatterplot of β versus α for the hurricane event time data.

Summary statistics are given below.

```
  summary(alpha.star)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2214  0.4830  0.5463  0.5501  0.6130  0.9767
  summary(beta.star)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.08241 0.86944 1.15283 1.25528 1.52438 5.42604
```

Note that the posterior mean for β is similar to the previously fixed value of β=1, and the posterior distribution for α is similar to that shown in Section 1. The correlation between α and β was estimated to be 0.22 and Figure 3c is a scatterplot of β versus α.

6 MCMC diagnostics

Inference using MCMC output is based on the sequence of samples $B+1$, $B+2$, …, $B+N$. Two reasonable questions then are

- What should B (burn-in length) be?
- How large should N, or really $N-B$ be?

As mentioned previously (Lecture Notes 8), some argue that discarding initial samples, the burn-in B, is wasteful and that a long enough chain (N) compensates for initial lack of convergence to the target distribution. We will not enter into that debate here, however, and simply note that that is a minority opinion—most users of MCMC output do discard initial chain values.

6.1 Burn-in

The ideal burn-in length B is the point at which the chain has run long enough that the resulting sample distribution equals the limiting distribution (the target distribution), i.e., the chain has converged.

Here we will discuss a couple of commonly used methods for specifying burn-in length. We note that none of these methods, however, *prove* that the chain has reached its limiting distribution. Instead the methods, more or less, help determine *if* the chain has *not* reached its limiting distribution.

6.1.1 Trace plots/Sample paths

The simplest method for specifying a burn-in length is to create a "time series" plot of the generated values, what is called a trace plot, as in Figures 1a, 2a, and 2b. These particular trace plots have been thinned (to reduce file size) but there is no evidence for any kind of pattern of being "stuck" in a certain region of the state space.

"Rapid" *apparent* movement about the state space is called *good mixing*.

6.1.2 Multiple Chains

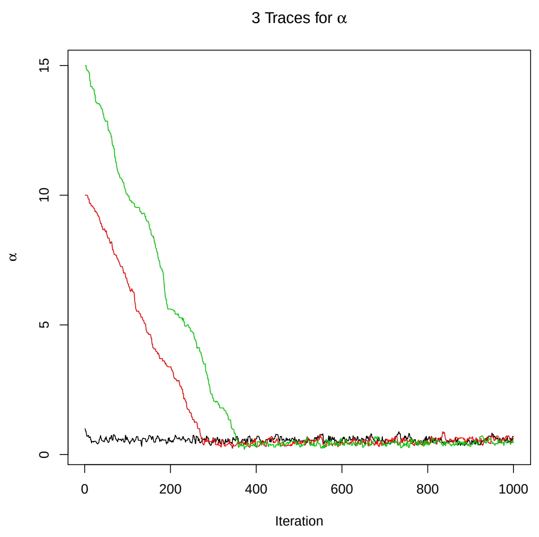
However, looking at a single chain's trace plots has limited value as it is possible that there are multiple modes in the posterior and only one mode is being sampled. A better approach is to run multiple chains with widely differing initial values and examine if and when the chains start to overlap. Figures 4a and 4b show the results for three chains for the hurricane data with initial values for α and β equal to 1, 10, and 15. The "convergence" of the α chains is considerably sooner than for the β parameter.

Again such plots do not prove convergence but they can indicate a lack of convergence. A burn-in pf 500 might be adequate for α while the burn-in for β might be at least 1,000. Sample values beyond these respective burn-in values do suggest "good mixing".

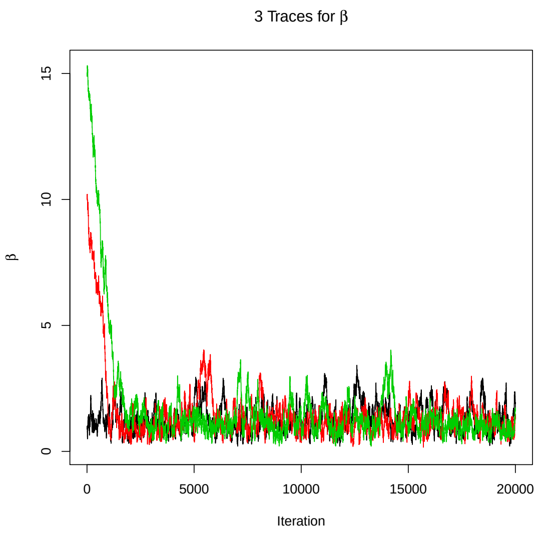
A more quantitative assessment of mixing (or lack of) is the Brooks-Gelman-Rubin (BGR) statistic. The idea is to compare the variability in the simulated values within each chain to the variability between chains. When chains have converged the variation within and between chains should be quite similar. The BGR statistic, denoted R, is defined as follows:

$$R = \frac{\text{Width of 80\% credible interval of all chains combined}}{\text{Average width of 80\% credible intervals for the individual chains}}$$

If the chain has converged, then R should be around 1. If the chains have not converged, then the width of the interval based on all chains combined will be greater than 1, as the between chain variation will be "large". The underlying principle is similar to that of an F-statistic in an analysis of variance (ANOVA) of K treatments, where K corresponds to the number of chains.



(a) Trace plots for α based on three different initial values, $\alpha^0 = 1$, 10, and 15.



(b) Trace plots for β based on three different initial values, $\beta^0 = 1$, 10, and 15.

A R Code

A.1 Metropolis-Hastings algorithm applied to hurricane event times data (α)

```
#-- Metropolis Hastings applied to the time between hurricanes data set
# with Weibull sampling distribution and Gamma prior for alpha
hurricane.gaps <- c(0.30, 4.61, 5.75, 0.24, 0.09, 0.18, 7.38, 1.20, 2.40, 0.18,
                    0.02, 10.07, 0.23, 0.44, 3.34, 0.06, 0.01, 0.71, 0.06, 0.42)
n <- length(hurricane.gaps)

set.seed(730)
N          <- 100000
kappa      <- lambda <- 0.1 #parameters for prior
a.q        <- 2             #parameters for proposal
b.q        <- 3
alpha.star <- numeric(N)
alpha.star[1] <- 1          # initial value

#--- To lessen change of underflows/overflows with MHR
# will log and then exponentiate

tally         <- 1
number.accept <- 0
while(tally < N) {
  tally <- tally+1
  # Proposal is a Gamma(a.q,b.q)
  candidate <- rgamma(n=1,shape=a.q,rate=b.q)
  previous  <- alpha.star[tally-1]
  log.joint.cand <- (n+kappa-1)*log(candidate) -  sum(hurricane.gaps^candidate) -
                    lambda*candidate +   (candidate-1)*sum(log(hurricane.gaps))
  log.joint.prev <- (n+kappa-1)*log(previous) -  sum(hurricane.gaps^previous) -
                    lambda*previous +   (previous-1)*sum(log(hurricane.gaps))
  log.q.cand     <- dgamma(candidate,a.q,b.q,log=TRUE)
  log.q.prev     <- dgamma(previous,a.q,b.q,log=TRUE)
  MHR            <- exp(log.joint.cand + log.q.prev - log.joint.prev - log.q.cand)
  u.star         <- runif(1)
  if(u.star <= min(1,MHR)) {
    # keep the candidate
    alpha.star[tally] <- candidate
    number.accept <- number.accept+1
  } else {
    # reject the candidate
    alpha.star[tally] <- previous
  }
}

cat("Acceptance rate=",round(number.accept/N,3),"\n")
# Acceptance rate= 0.262
#thin for plotting to reduce file size
plot(alpha.star[seq(1,N,length=2000)],type="l",
     xlab="Iteration",ylab=expression(alpha),main="(thinned) Markov chain")
plot(density(alpha.star),type="l",xlab=expression(alpha),
     main=expression(paste("Posterior Distribution for ",alpha)))

Burnin <- 1000
alpha.star <- alpha.star[-c(1:Burnin)]
summary(alpha.star)
#  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.2443  0.4919  0.5489  0.5512  0.6097  0.9477

cat("Probability alpha between 0.40 and 0.71=",
    sum(alpha.star>=0.40 & alpha.star<=0.71)/(N-Burnin),"\n")
# Probability alpha between 0.40 and 0.71= 0.9257273
```

A.2 MH joint update of α and β for hurricane data

```
#-- Metropolis-Hastings updating 2 parameters: first will do joint updating.
hurricane.gaps <- c(0.30, 4.61, 5.75, 0.24, 0.09, 0.18, 7.38, 1.20, 2.40, 0.18,
                    0.02, 10.07, 0.23, 0.44, 3.34, 0.06, 0.01, 0.71, 0.06, 0.42)
n <- length(hurricane.gaps)

set.seed(730)
N           <- 100000
kappa       <- lambda <- 0.1 #parameters for priors
sigma.alpha <- 0.1          #parameters for proposal
sigma.beta  <- 0.1
alpha.star <- beta.star <- numeric(N)
alpha.star[1] <- 1          # initial values
beta.star[1]  <- 1

#--- To lessen change of underflows/overflows with MHR
# will log and then exponentiate

y <- hurricane.gaps

tally         <- 1
number.accept <- 0
while(tally < N) {
  tally <- tally+1
  # Proposals are Normal(alpha^{-1},sigma.alpha), Normal(beta^{-1},sigma.beta), thus
  # symmetric (random walk)
  alpha.c  <- rnorm(n=1,mean=alpha.star[tally-1],sd=sigma.alpha)
  beta.c   <- rnorm(n=1,mean=beta.star[tally-1],sd=sigma.beta)

  if(alpha.c < 0 || beta.c < 0) {
     alpha.star[tally] <- alpha.star[tally-1]
     beta.star[tally]  <- beta.star[tally-1]

  } else {

  alpha.p <- alpha.star[tally-1]
  beta.p  <- beta.star[tally-1]

  log.joint.cand <- (kappa-1)*log(alpha.c*beta.c)-lambda*(alpha.c+beta.c) +
     n*log(alpha.c/beta.c) + (alpha.c-1)*sum(log(y/beta.c)) -
     sum((y/beta.c)^alpha.c)

  log.joint.prev <- (kappa-1)*log(alpha.p*beta.p)-lambda*(alpha.p+beta.p) +
  n*log(alpha.p/beta.p) + (alpha.p-1)*sum(log(y/beta.p)) -
  sum((y/beta.p)^alpha.p)

  MHR            <- exp(log.joint.cand - log.joint.prev)
  u.star         <- runif(1)
  if(u.star <= min(1,MHR)) {
    # keep the candidate
    alpha.star[tally] <- alpha.c
    beta.star[tally]  <- beta.c
    number.accept <- number.accept+1
  } else {
    # reject the candidate
    alpha.star[tally] <- alpha.p
    beta.star[tally]  <- beta.p
   }
  }
}
```

```
cat("Acceptance rate=",round(number.accept/N,3),"\n")
# Acceptance rate= 0.658

#Trace plots and densities
thin.N <- 10000
plot(alpha.star[seq(1,N,length=thin.N)],type="l",
     xlab="Iteration",ylab=expression(alpha),main=expression(paste("Trace for ",alpha)))

plot(density(alpha.star),type="l",xlab=expression(alpha),
     main=expression(paste("Posterior Distribution for ",alpha)))

plot(beta.star[seq(1,N,length=thin.N)],type="l",
     xlab="Iteration",ylab=expression(alpha),main=expression(paste("Trace for ",beta)))

plot(density(beta.star),type="l",xlab=expression(beta),
     main=expression(paste("Posterior Distribution for ",beta)))

# distributional summaries
Burnin <- 1000
alpha.star <- alpha.star[-c(1:Burnin)]
beta.star  <- beta.star[-c(1:Burnin)]
summary(alpha.star)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#  0.2214  0.4830  0.5463  0.5501  0.6130  0.9767
summary(beta.star)
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#  0.08241 0.86944 1.15283 1.25528 1.52438 5.42604
```

November 14, 2020