L10: More MCMC Diagnostics & Gibbs Sampling

1 MCMC diagnostics: How Large a Sample?

Once the burn-in length, B, is determined, there is still the question of how much longer the chain should be run. Denote the number of iterations beyond B by N. Thus the total chain length is $B + N$.

Ultimately, the size of N depends upon the desired precision of the estimate, the "Monte Carlo error" of the estimate.

We'll consider the following estimator:

$$\hat{E}(\theta) = \frac{1}{N} \sum_{j=B+1}^{N+B} \theta^j$$

To reduce notation, let n= N and let i=$j - B$, so the indexing i starts at 1:

$$\hat{E}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \theta^i \tag{1}$$

1.1 Variance of $\hat{E}(\theta)$

Again, remember that we are measuring the Monte Carlo error of the estimate $\hat{E}(\theta)$. This is **NOT** the variance of θ, nor the variance of $E[\theta]$, a constant, which is then 0.

In the following we are assuming we have a stationary Markov chain, in particular that correlation between realizations at "times" i and j in the chain, $\text{corr}(\theta_i, \theta_j)$ depends only on the absolute difference between i and j, $|i - j|$.

Notation:

- γ_k=$Cov(\theta^i, \theta^{i+k})$, is the autocovariance of lag k ($k \geq 0$) of the chain
- σ^2=γ_0, the variance of θ^t
- $\rho_k = \gamma_k/\sigma^2$, the autocorrelation of lag k
- τ_n^2/n, the variance of $\hat{E}(\theta)$ (details below)

The variance of $\hat{E}(\theta)$:

$$Var\left[\hat{E}(\theta)\right] = Var\left[\frac{1}{n}\sum_{i=1}^{n}\theta^i\right] = \frac{1}{n^2}\left[\sum_{i=1}^{n} Var(\theta^i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} Cov(\theta^i, \theta^j)\right] = \frac{1}{n^2}\left[n\sigma^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\gamma_{j-i}\right]$$

$$= \frac{1}{n^2}\left[n\sigma^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sigma^2\rho_{j-i}\right] = \frac{1}{n^2}\left[n\sigma^2 + 2\sigma^2\sum_{k=1}^{n}(n-k)\rho_k\right] = \frac{\sigma^2}{n}\left[1 + 2\sum_{k=1}^{n}\frac{n-k}{n}\rho_k\right] = \frac{\tau_n^2}{n}$$

where

$$\tau_n^2 = \sigma^2\left[1 + 2\sum_{k=1}^{n}\frac{n-k}{n}\rho_k\right]$$

The following is *not* a rigorous proof. We are assuming that ρ_k is decreasing as k increases, in particular $\sum_{k=1}^{\infty} \rho_k$ is finite. As n gets "large"

$$\tau_n^2 = \sigma^2 \left[1 + 2 \sum_{k=1}^{n} \frac{n-k}{n} \rho_k \right] \approx \sigma^2 \left(1 + 2 \sum_{k=1}^{\infty} \rho_k \right) \tag{2}$$

Thus

$$V(\hat{E}(\theta)) \approx \frac{\sigma^2 \left(1 + 2 \sum_{k=1}^{\infty} \rho_k\right)}{n} \tag{3}$$

And as usual, the variance, a measure of the Monte Carlo error, of $\hat{E}(\theta)$ can be reduced by increasing n, but that variance is a function of the autocorrelation, too.

1.2 Effective Sample Size, ESS

"The term $1 + 2 \sum_{k=1}^{\infty} \rho_k$ is called the *inefficiency factor* or *integrated autocorrelation* because it measures how far the θ^i's are from being a random sample and how much $V(\hat{E}(\theta))$ increases because of that"[1]. From this the *effective sample size*, n_{eff}, can be calculated:

$$n_{eff} = \frac{n}{1 + 2 \sum_{k=1}^{\infty} \rho_k} \tag{4}$$

Then

$$V(\hat{E}(\theta)) \approx \frac{\sigma^2}{n_{eff}} \tag{5}$$

A 2018 paper by Elvira, et al[2] is a critical examination of the notion of ESS for importance sampling, thus independent samples, and I just want to add a cautionary note that procedures for estimating ESS may change in the future.

1.3 Autocorrelation plots

Thus if the autocorrelation is positive and large, then the variance of $V(\hat{E}(\theta))$ is large relative to the variance of $\hat{E}(\theta)$ based on *independent* samples of θ^i, i=1,...,n. A useful diagnostic for assessing the potential severity of autocorrelation is an autocorrelation plot, a plot of $\hat{\rho}_k$ against lag k where[3]

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\sigma^2}} = \frac{\sum_{i=1}^{n-k} (\theta^i - \bar{\theta})(\theta^{i+k} - \bar{\theta})}{\sum_{i=1}^{n} (\theta^i - \bar{\theta})^2} \tag{6}$$

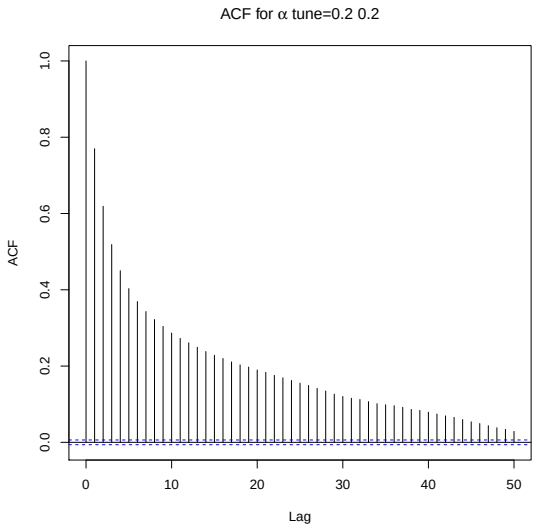
where $\bar{\theta}$ is the sample average.

Figure 1 shows the autocorrelation plots for the Weibull parameters α and β when they are jointly updated using a Normal random walk, $\theta^c \sim \text{Normal}(\theta^{t-1}, \sigma_\theta^2)$, where the "tuning parameters" σ_θ^2 were set equal to 0.2^2 for both parameters. The acceptance rate was 0.44 with these "tuning parameters". Reducing the acceptance rate by increasing the size of σ_α^2 and σ_β^2 has some effect on the degree of autocorrelation. For example increasing σ_α^2 to 0.5^2 and σ_β^2 to 1.2^2 lowered the acceptance rate to 0.12 but reduced the autocorrelation in both α and β (see Figure 2).

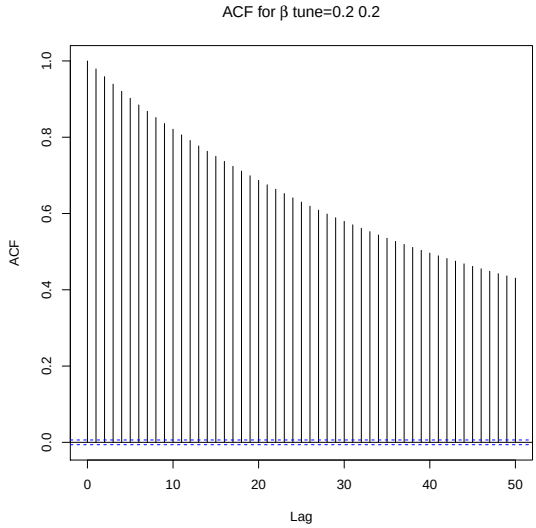
[1] *Markov Chain Monte Carlo, 2nd edition*, 2006, Gamerman and Lopes, p26.

[2] "Rethinking the Effective Sample Size" which can be found at https://arxiv.org/abs/1809.04129.

[3] From Time Series Analysis and Its Applications, 4th Ed, Shumway and Stoffer, 2017, $\hat{\rho}_k = \gamma(k)/\gamma(0)$, where $\gamma(k) = \frac{1}{n} \sum_{i=1}^{n-k} (\theta^i - \bar{\theta})(\theta^{i+k} - \bar{\theta})$. Thus the divisor n cancels in numerator and denominator to yield eq'n 6.

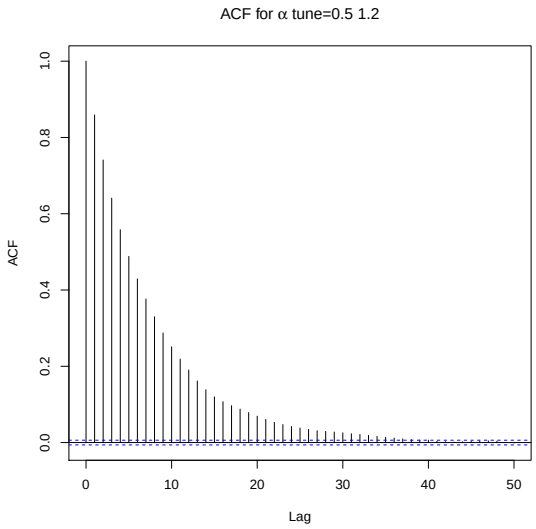


(a) Autocorrelation for α.

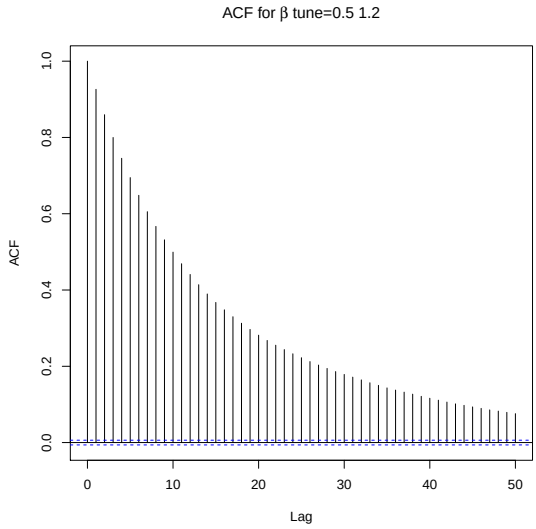


(b) Autocorrelation for β.

Figure 1: Autocorrelations for α and β with the Weibull model for hurricane event times. Normal random walk proposals used for α and β where σ_α^2=0.2^2 and σ_β^2=0.2^2.



(a) Autocorrelation for α.



(b) Autocorrelation for β.

Figure 2: Autocorrelation for α and β with the Weibull model for hurricane event times. Normal random walk proposals used for α and β where σ_α^2=0.4^2 and σ_β^2=1.2^2.

1.4 Central limit theorem for $\hat{E}(\theta)$

Under certain conditions on the Markov chain (that it be uniformly ergodic) and if $E[\theta^2]$ exists, then

$$\hat{E}(\theta) \sim \text{Asymptotically Normal}\left(E[\theta], \frac{\sigma^2}{n_{eff}}\right)$$

where by Asymptotically Normal we mean:

$$\frac{\hat{E}(\theta) - E(\theta)}{\sigma/\sqrt{n_{eff}}} \overset{distribution}{\rightarrow} \text{Normal}(0,1)$$

This then allows us to calculate approximate interval estimates for $E[\theta]$ using the normal distribution. However, the main difficulty is estimating the variance, σ^2/n_{eff}.

1.5 Estimating $V(\hat{E}(\theta))$

Due to the autocorrelation, estimation of $V[\hat{E}(\theta)]$ is not a trivial matter. A popular method for calculating the variance is *batching*.

The algorithm for batching:

- Partition the chain of length n into m batches of T successive values.
- For each batch calculate the average value of θ within the batch. In batch i

$$\bar{\theta}_i = \frac{1}{T}\sum_{j=1}^{T}\theta_{i,j}, i = 1, \ldots, m$$

- The estimate of $\frac{\tau^2}{n}$ is then:

$$\frac{\widehat{\tau^2}}{n} = \frac{\frac{T}{m-1}\sum_{i=1}^{m}(\bar{\theta}_i - \bar{\theta})^2}{n}$$

"Values of T should be between 10 and 30" (Gamerman and Lopes, 2006, p 134).

The `R` package `coda` includes a function called `batchSE` that will calculate the standard errors with this method. If you run the code in Appendix A.2 of Lecture Notes 9, with the tuning parameters of σ_α=0.5 and σ_β=1.2:

```
library(coda)
#Burnin=1000, N=100,000
a.b.star <- mcmc(data=cbind(alpha.star,beta.star),start=Burnin,end=N)
MonteCarlo.se <- batchSE(x=a.b.star,batchSize=round(N/30))
print(MonteCarlo.se)
#  alpha.star   beta.star
# 0.001151362 0.012550150
```

The package `coda` will also calculate effective sample size, n_{eff}, with the function. For example,

```
effectiveSize(a.b.star)
# alpha.star  beta.star
# 6835.639   3130.651
```

The effective sample sizes are relatively low given N=100,000, roughly 6% and 3% for α and β due to the relatively high autocorrelation.

Note: Sometimes individuals will thin the chain, say keep every 100th value, to reduce autocorrelation. This does *not* reduce Monte Carlo error. However, it can be useful for reducing file sizes.

1.6 Improving performance

Monte Carlo error is exacerbated by slow mixing, relatively low acceptance rates, high autocorrelation, and generally limited or poor coverage of the state space. Potential remedies include:

1. Changing the proposal distribution. With random walk proposals one can change the variance of the proposal distribution. For example, the random walk sampler for the Weibull problem was:

$$\alpha^c \sim \text{Normal}(\alpha^t, \sigma_\alpha^2)$$
$$\beta^c \sim \text{Normal}(\beta^t, \sigma_\beta^2)$$

   Changing the values of σ_α^2 and σ_β^2 affected the autocorrelation (and the acceptance rates). Such manipulation of proposals is an example of *tuning the proposals*. This is a completely legitimate exercise, it is not "cheating" in any way as the prior distributions do not change. It's just a technique to increase effective sample size.

2. Reparameterising the model. For example, with Bayesian regression models, centering the covariates can reduce autocorrelation. For example, compare the following two parameterisations of a two covariate multiple regression:

$$\text{Formulation 1: } y \sim \text{Normal}\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2, \sigma^2\right)$$
$$\text{Formulation 2: } y \sim \text{Normal}\left(\beta_0 + \beta_1 (x_1 - \bar{x}_1) + \beta_2 (x_2 - \bar{x}_2), \sigma^2\right)$$

   Formulation 2 will generally lead to lower autocorrelation in the simulated values of β_0, β_1, and β_2.

3. Blocking: when parameters have some degree of positive correlation, it is often better to jointly update such parameters. For example if there are 4 parameters where θ_1 and θ_2 are highly correlated, then it may be better to generate (θ_1^c, θ_2^c) simultaneously to increase the acceptance rate, improve mixing, increase effective sample size.

2 Gibbs Sampler

The Gibbs Sampling is another very popular MCMC method for generating samples from multivariate distributions. While it can be shown to be a special case of a Metropolis-Hastings sampler, it has two unique features:

1. The proposal distributions are the conditional distributions for the θ's.
2. All candidate values are accepted.

2.1 The algorithm

The algorithm as described herein will closely parallel the single update Metropolis-Hastings procedure described previously.

Let $\Theta = (\theta_1, \theta_2, \ldots, \theta_q)$ with distribution $p(\Theta)$.

Denote the full conditional distribution for θ_i by

$$p(\theta_i|\theta_{(-i)}) = p(\theta_i|\theta_1, \theta_2, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_q), \ i = 1, 2, \ldots, q$$

Initialise the chain with $\Theta^0 = (\theta_1^0, \theta_2^0, \ldots, \theta_q^0)$.

At iteration $t+1$ given the values Θ^t, generate Θ^{t+1} as follows:

- Generate θ_1^{t+1} from $p(\theta_1|\theta_2^t, \theta_3^t, \ldots, \theta_q^t)$
- Generate θ_2^{t+1} from $p(\theta_2|\theta_1^{t+1}, \theta_3^t, \ldots, \theta_q^t)$
- ...
- Generate θ_i^{t+1} from $p(\theta_i|\theta_1^{t+1}, \theta_2^{t+1}, \ldots, \theta_{i-1}^{t+1}, \theta_{i+1}^t, \ldots, \theta_q^t)$
- ...
- Generate θ_q^{t+1} from $p(\theta_q|\theta_1^{t+1}, \theta_2^{t+1}, \ldots, \theta_{q-1}^{t+1})$

The (joint) transition probability of going from Θ^t to Θ^{t+1} is then

$$\mathcal{K}_{Gibbs}(\Theta^t, \Theta^{t+1}) = \prod_{i=1}^{q} p(\theta_i^{t+1}|\theta_j^{t+1}, j < i, \text{ and } \theta_j^t, j > i)$$

and the stationary distribution is the target distribution $p(\Theta)$.

2.2 Example 1: Bivariate Normal with known correlation

The Gibbs Sampler is demonstrated for a two-dimensional parameter, $\Theta=(\theta_1, \theta_2)$, where the target distribution is a "standard" bivariate normal with means equal to 0, standard deviations equal to 1, and known correlation ρ:

$$\theta_1, \theta_2 \sim \text{BVN}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$$

The joint pdf is:

$$p(\theta_1, \theta_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{\theta_1^2 - 2\rho\theta_1\theta_2 + \theta_2^2}{2(1-\rho^2)}\right)$$

To find the conditional distribution for θ_1 given θ_2, focus on just those terms in the joint distribution involving θ_1:

$$p(\theta_1|\theta_2) \propto \exp\left(-\frac{\theta_1^2 - 2\rho\theta_1\theta_2}{2(1-\rho^2)}\right) \propto \exp\left(-\frac{(\theta_1 - \rho\theta_2)^2}{2(1-\rho^2)}\right)$$

which can be seen to be the kernel of a Normal($\rho\theta_2$, $1-\rho^2$) density function. Similarly, $\theta_2|\theta_1 \sim$ Normal($\rho\theta_1, 1-\rho^2$).

The Gibbs Sampler: Specify initial values (θ_1^0, θ_2^0). At iteration $t+1$,

1. Generate $\theta_1^{t+1}|\theta_2^t$ from Normal($\rho\theta_2^t, (1-\rho^2)$).
2. Then generate $\theta_2^{t+1}|\theta_1^{t+1}$ from Normal($\rho\theta_1^{t+1}, (1-\rho^2)$).

Example `R` code (additional code in Appendix A.1):

```
rho <- 0.2
N <- 10000
theta1.star <- theta2.star <- numeric(N)
theta1.star[1] <- 1.3
theta2.star[1] <- -0.2
for(i in 2:N) {
  theta1.star[i] <- rnorm(n=1,mean=rho*theta2.star[i-1],sd=sqrt(1-rho^2))
  theta2.star[i] <- rnorm(n=1,mean=rho*theta1.star[i],sd=sqrt(1-rho^2))
}
```

The Monte Carlo errors and effective sample sizes, are shown below (Note: N=10,000).

	θ_1	θ_2
MC error	0.01204288	0.00953518
n_{eff}	8647	8041

The trace plots and density plots for the generated sample are shown in Figure 3. The joint distribution of θ_1 and θ_2 and the autocorrelation for θ_1 are plotted in Figure 4. The trace plots, low autocorrelation, Monte Carlo errors, and large effective sample sizes (over 80% of N) all indicate good mixing.

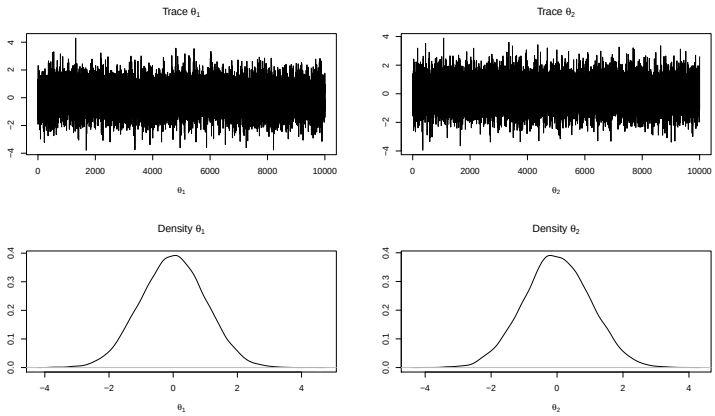
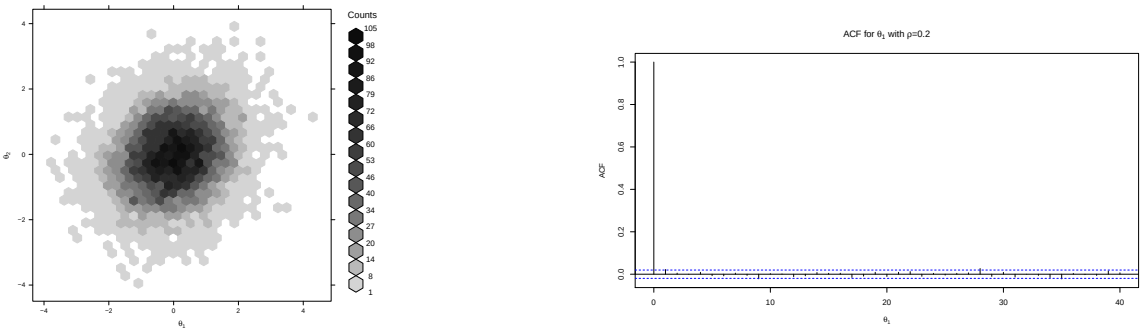


Figure 3: Trace and density plots for Gibbs Sampler values of θ_1 and θ_2 from a standard bivariate normal with ρ=0.2



(a) Joint distribution.

(b) ACF.

Figure 4: Joint distribution for Gibbs Sampler values of θ_1 and θ_2 from a standard bivariate normal with ρ=0.2.

By way of contrast the above exercise was repeated with a strong negative correlation, ρ= -0.95. The trace and density plots are shown in Figure 5 and a pattern in the trace plots can be seen. The joint density and the autocorrelation for θ_1 are shown in Figure 6 and make clear the effects of the high correlation. Monte Carlo errors are considerably larger and the effective sample sizes considerably lower:

	θ_1	θ_2
MC error	0.0526	0.0521
n_{eff}	447	467

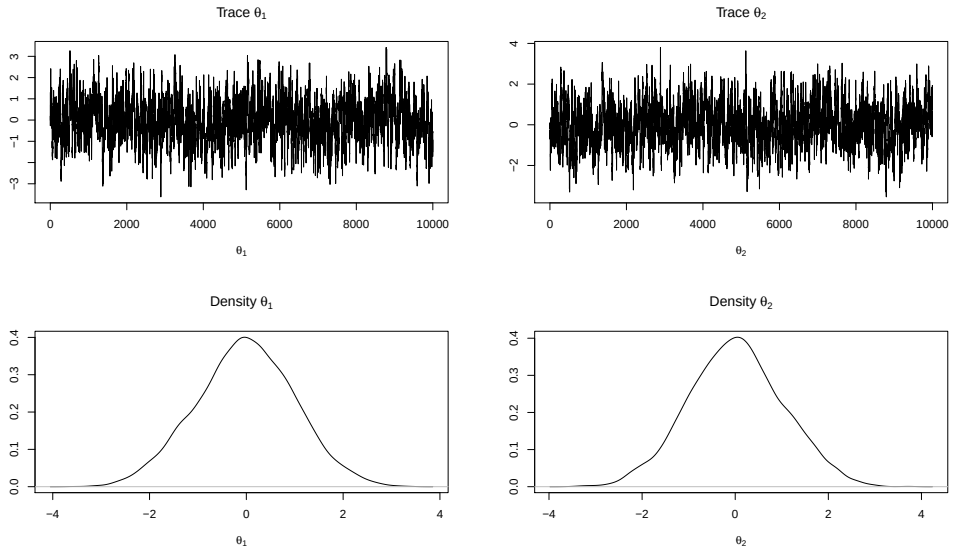


Figure 5: Trace and density plots for Gibbs Sampler values of θ_1 and θ_2 from a standard bivariate normal with ρ= -0.95.

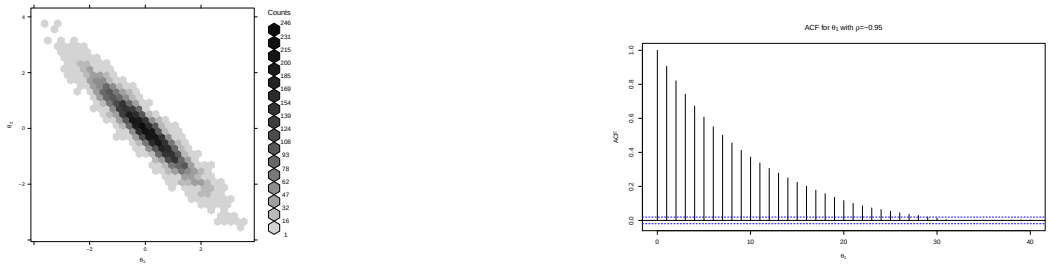


Figure 6: Joint distribution for Gibbs Sampler values of θ_1 and θ_2 from a standard bivariate normal with ρ= -0.95 and autocorrelation for θ_1.

2.3 A special case of Metropolis-Hastings

Consider the Metropolis-Hastings algorithm with single updates using the conditional distribution as the proposal. As a concrete example let q=3, thus $\Theta = (\theta_1, \theta_2, \theta_3)$, and let the updating be in the order 1, 2, and then 3. At the beginning of iteration $t+1$ we have $\Theta^t = (\theta_1^t, \theta_2^t, \theta_3^t)$. The "candidate" value for θ_1 is generated from $p(\theta_1|\theta_2^t, \theta_3^t)$, and the Metropolis Hastings ratio is then:

$$MHR(\theta_1^t, \theta_1^c) = \frac{p(\theta_1^c, \theta_2^t, \theta_3^t)}{p(\theta_1^t, \theta_2^t, \theta_3^t)} \frac{q(\theta_1^t|\theta_1^c, \theta_2^t, \theta_3^t)}{q(\theta_1^c|\theta_1^t, \theta_2^t, \theta_3^t)} \tag{7}$$

$$= \frac{p(\theta_1^c|\theta_2^t, \theta_3^t)p(\theta_2^t, \theta_3^t)}{p(\theta_1^t|\theta_2^t, \theta_3^t)p(\theta_2^t, \theta_3^t)} \frac{p(\theta_1^t|\theta_2^t, \theta_3^t)}{p(\theta_1^c|\theta_2^t, \theta_3^t)} \tag{8}$$

$$= \frac{p(\theta_2^t, \theta_3^t)}{p(\theta_2^t, \theta_3^t)} = 1 \tag{9}$$

Thus the candidate value is always accepted.

2.4 Gibbs Sampler vs (General) Metropolis-Hastings

- Gibbs Sampler Strengths
  - The proposal distribution is automatically defined (thus no "tuning" of a proposal)
  - Always "accept" the candidate value
  - Because the proposal is conditional on the other parameters, can view it as an "adaptive" algorithm as changes in one subset of parameters likely lead to changes in another subset of parameters
- Gibbs Sampler Weaknesses
  - The conditional distributions may not be tractable
  - Relatedly, sampling from the conditionals may be computationally intensive
  - 100% acceptance rate does not necessarily mean good mixing
- Metropolis-Hastings Strengths
  - Proposal distribution quite flexible, can be fast to sample from and fast to evaluate MHR (particularly with symmetric proposals)
  - Don't need to know the conditional distributions
  - Block updating can be relatively easy; e.g., multivariate t distribution as a proposal.[4].
- Metropolis-Hastings Weaknesses
  - May be hard to find a good proposal, one with good mixing
  - Can take time to "tune" a proposal, e.g., a good variance value for a normal random walk proposal

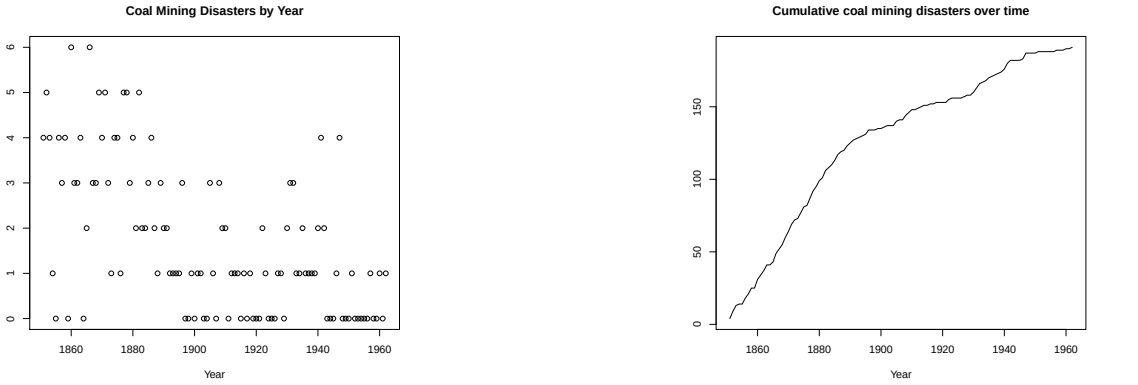
2.5 Combining Gibbs Sampling and Metropolis-Hastings

Given $\Theta = (\theta_1, \theta_2, \ldots, \theta_q)$, if full conditionals for all θ_i, i=1, ..., q are not tractable, then Metropolis-Hastings can be used in those cases. For example, suppose $\Theta = (\theta_1, \theta_2, \theta_3)$ and $p(\theta_1|\theta_2, \theta_3)$ and $p(\theta_2|\theta_1, \theta)$ are tractable, but $p(\theta_3|\theta_1, \theta_2)$ is not. Then the Gibbs Sampler can be used to generate θ_1 and θ_2 and Metropolis-Hastings, e.g. a random walk proposal, can be used to generate θ_3. Of course, θ_1 and θ_2 will have 100% acceptance rates, but θ_3 will not.

[4]Block updating can be done with the Gibbs sampler, too, so long as the conditional distribution for the "block" is tractable. For example, generate (θ_1, θ_2) simultaneously with $p(\theta_1, \theta_2|\theta_3, \ldots, \theta_q)$.

2.6 Example 2: Coal mining disasters with change point

This example[5] models the occurrence of coal mining disasters (with 10 or more fatalities) in Great Britain from the years 1851 through 1962 (n=112)[6]. Figure 7 shows the occurrences by year and the cumulative number of disasters over time.



(a) Number of occurrences vs year.

(b) Cumulative disasters over time.

Figure 7: Occurrences of coal mining disasters in Great Britain.

A Poisson distribution is a commonly used distribution for such count data. However as the plots in Figure 7 indicate there appears to be a reduction in the number of disasters some years before 1900, thus a single Poisson distribution does not seem adequate. A *change point* model is one means of accounting for a potential change in the incidence rates with one Poisson distribution before the change and a different Poisson distribution afterwards. Let y_i denote the number of disasters in year i and let τ indicate the last year that the first Poisson occurred, where τ could equal 1, ..., $n-1$. Assume that the distribution of y_i, i=1,..., τ is Poisson(μ_b) and y_i, i=$\tau+1$, ..., n is Poisson(μ_a), where b and a denote before and after.

Specify Gamma distribution priors for the Poisson parameters: $\mu_b \sim \text{Gamma}(\alpha, \beta)$ and $\mu_a \sim \text{Gamma}(\gamma, \delta)$. Assume that any of the change point years are equally likely, thus τ is uniformly distributed over 1,2,..., $n-1$, or $\Pr(\tau = i) = 1/n-1$, where n=112.

The posterior distribution for μ_b, μ_a, and τ, letting $\boldsymbol{y}$= $(y_1, \ldots, y_n)$:

$$\begin{aligned}
p(\mu_b, \mu_a, \tau|\boldsymbol{y}) &\propto \pi(\mu_b)\pi(\mu_a)\pi(\tau)f(\boldsymbol{y}|\mu_b, \mu_a, \tau) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\mu_b^{\alpha-1}e^{-\mu_b\beta}\frac{\delta^\gamma}{\Gamma(\gamma)}\mu_a^{\gamma-1}e^{-\mu_a\delta}\frac{1}{n}\prod_{i=1}^{\tau}\frac{e^{-\mu_b y_i}{\mu_b}^{y_i}}{y_i!} \times \prod_{i=\tau+1}^{n}\frac{e^{-\mu_a y_i}{\mu_a}^{y_i}}{y_i!} \\
&\propto \mu_b^{\alpha-1}e^{-\mu_b\beta}\mu_a^{\gamma-1}e^{-\mu_a\delta}\prod_{i=1}^{\tau}e^{-\mu_b y_i}{\mu_b}^{y_i} \times \prod_{i=\tau+1}^{n}e^{-\mu_a y_i}{\mu_a}^{y_i} \\
&= \mu_b^{\alpha+s_\tau-1}e^{-(\beta+\tau)\mu_b}\mu_a^{\gamma+s_n-s_\tau-1}e^{-(\delta+n-\tau)\mu_a}
\end{aligned}$$

where $s_\tau = \sum_{i=1}^{\tau} y_i$ and $s_n = \sum_{i=1}^{n} y_i$.

The "trick" to finding the conditional for a parameter θ is to just consider terms including θ. Doing that for

[5]Taken from Markov Chain Monte Carlo (2006) by Gamerman and Lopes, pp 143–146).

[6]These data can be accessed at https://conservancy.umn.edu/handle/11299/200478.

μ_b, μ_a, and τ:

$$p(\mu_b|\mu_a, \tau, \boldsymbol{y}) \propto \mu_b^{\alpha+s_\tau-1} e^{-(\beta+\tau)\mu_b} \tag{10}$$

$$p(\mu_a|\mu_b, \tau, \boldsymbol{y}) \propto \mu_a^{\gamma+s_n-s_\tau-1} e^{-(\delta+n-\tau)\mu_a} \tag{11}$$

$$p(\tau|\mu_a, \mu_b, \boldsymbol{y}) \propto \mu_b^{\alpha+s_\tau-1} e^{-(\beta+\tau)\mu_b} \mu_a^{\gamma+s_n-s_\tau-1} e^{-(\delta+n-\tau)\mu_a} \tag{12}$$

For the posterior conditional of μ_b, the righthand side of (10) is the kernel of a Gamma($\alpha + s_\tau$, $\beta + \tau$), while the righthand side of (11) is the kernel of a Gamma($\gamma + s_n - s_\tau$, $\delta + n - \tau$). For τ the conditional distribution is not a standard form but because there are a finite number of values for τ (1, 2, ..., $n-1$) and the sum of the conditionals over all possible values must sum to 1, the posterior for τ is then a discrete probability distribution:

$$p(\tau|\mu_a, \mu_b, \boldsymbol{y}) = \frac{\mu_b^{\alpha+s_\tau-1} e^{-(\beta+\tau)\mu_b} \mu_a^{\gamma+s_n-s_\tau-1} e^{-(\delta+n-\tau)\mu_a}}{\sum_{\tau=1}^{n-1} \mu_b^{\alpha+s_\tau-1} e^{-(\beta+\tau)\mu_b} \mu_a^{\gamma+s_n-s_\tau-1} e^{-(\delta+n-\tau)\mu_a}} \tag{13}$$

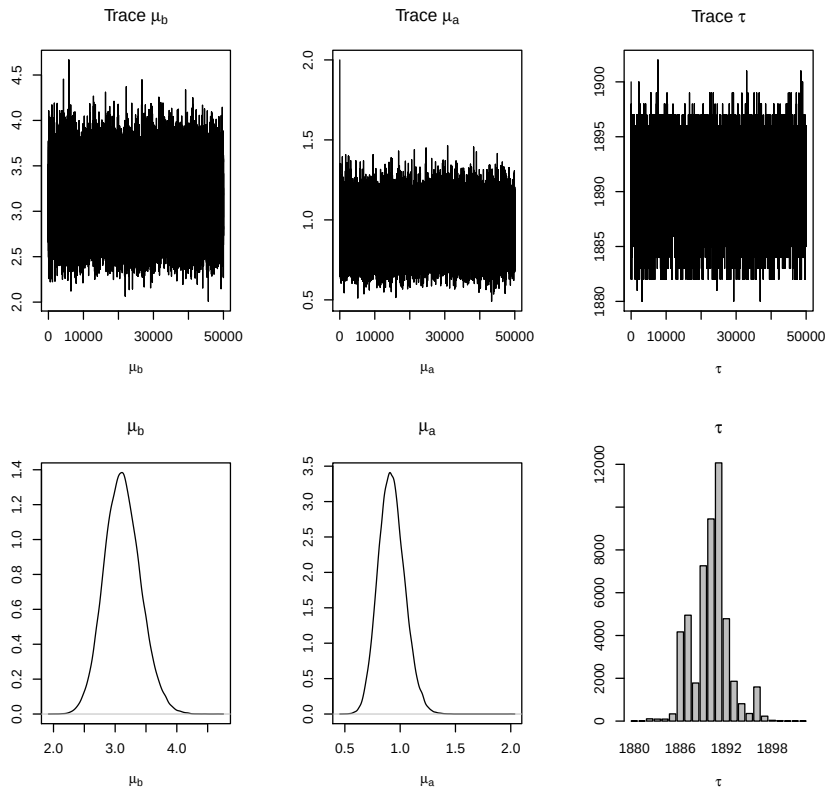
Setting the hyperparameters of the priors for μ_b, α and β, and for μ_a, γ and δ, all equal 0.001, the Gibbs sampler was implemented with the following R code.

```
N <- 50000
y <- coal.data$Disasters
s.n <- sum(y)
n <- length(y)
cum.y <- cumsum(y[-n])
tau.vec <- 1:(n-1)

alpha.b <- beta.b <- gamma.a <- delta.a <- 0.001

mub.star <- mua.star <- tau.star <- numeric(N)
mub.star[1] <- 3; mua.star[1] <- 2; tau.star[1] <- 50

for(i in 2:N) {
  s.tau <- sum(y[1:tau.star[i-1]])
  mub.star[i] <- rgamma(n=1,alpha.b+s.tau,beta.b+tau.star[i-1])

  mua.star[i] <- rgamma(n=1,gamma.a+s.n-s.tau,delta.a+n-tau.star[i-1])

  temp <-        (alpha.b+cum.y-1)*log(mub.star[i]) -(beta.b+tau.vec)*mub.star[i] +
             (gamma.a+s.n-cum.y-1)*log(mua.star[i]) -(delta.a+n-tau.vec)*mua.star[i]
  prob.vector <- exp(temp)
  prob.vector <- prob.vector/sum(prob.vector)
  tau.star[i] <- sample(x=1:(n-1),size=1,prob=prob.vector)
}
```

The trace plots and estimated posterior distributions are shown in Figure 8. Summary statistics are shown below. The year 1890 is the expected year of a change.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
μ_b	2.01	2.92	3.11	3.12	3.31	4.45
μ_a	0.49	0.84	0.92	0.92	1.00	1.46
τ	1880.00	1889.00	1890.00	1889.97	1891.00	1901.00

Figure 8: Trace plots and estimated posterior distributions for parameters of change point Poisson model for occurrences of coal mining disasters in Great Britain.



A R Code

A.1 Gibbs Sampler for standard BVN with known ρ

```
#--- Gibbs Sampler ------------------------
# BVN with standard normal and known rho
set.seed(92)
rho <- 0.2
# rho <- -0.95   # the high negative correlation example
N <- 10000
theta1.star <- theta2.star <- numeric(N)
theta1.star[1] <- 1.3
theta2.star[1] <- -0.2
for(i in 2:N) {
  theta1.star[i] <- rnorm(n=1,mean=rho*theta2.star[i-1],sd=sqrt((1-rho^2)))
  theta2.star[i] <- rnorm(n=1,mean=rho*theta1.star[i],sd=sqrt((1-rho^2)))
}

# Trace and Density plots
par(mfrow=c(2,2),oma=c(0,0,3,0))
plot(1:N,theta1.star,xlab=expression(theta[1]),ylab="",main=expression(paste("Trace ",theta[1])),
     type="l")
plot(1:N,theta2.star,xlab=expression(theta[2]),ylab="",main=expression(paste("Trace ",theta[2])),
     type="l")
plot(density(theta1.star),xlab=expression(theta[1]),ylab="",
     main=expression(paste("Density ",theta[1])),type="l")
plot(density(theta2.star),xlab=expression(theta[2]),ylab="",
     main=expression(paste("Density ",theta[2])),type="l")
par(mfrow=c(1,1))

#-- Joint density
library(hexbin)
hexbinplot(theta2 ~ theta1,data=data.frame(theta1=theta1.star,theta2=theta2.star),
           xlab=expression(theta[1]),ylab=expression(theta[2]))

#-- examine autocorrelation
plot(acf(theta1.star),xlab=expression(theta[1]),
     main=bquote(paste("ACF for ",theta[1]," with ",rho,"=",.(rho))))

#-- examine Monte Carlo standard error and effective sample size
library(coda)
Burnin <- 1000
t1.t2.star <- mcmc(data=cbind(theta1.star,theta2.star),start=Burnin,end=N)
MonteCarlo.se <- batchSE(x=t1.t2.star,batchSize=round(N/30))
print(MonteCarlo.se)
#  theta1.star theta2.star
#   0.01204288  0.00953518

round(effectiveSize(t1.t2.star))
# theta1.star theta2.star
# 8647        8041
```

A.2 Gibbs sampler for coal mine disasters

```
set.seed(73)
N <- 50000
y <- coal.data$Disasters
s.n <- sum(y); n <- length(y)
cum.y <- cumsum(y[-n])
tau.vec <- 1:(n-1)

alpha.b <- beta.b <- gamma.a <- delta.a <- 0.001

mub.star <- mua.star <- tau.star <- numeric(N)
mub.star[1] <- 3; mua.star[1] <- 2; tau.star[1] <- 50

for(i in 2:N) {
  s.tau <- sum(y[1:tau.star[i-1]])
  mub.star[i] <- rgamma(n=1,alpha.b+s.tau,beta.b+tau.star[i-1])

  mua.star[i] <- rgamma(n=1,gamma.a+s.n-s.tau,delta.a+n-tau.star[i-1])

  temp <-        (alpha.b+cum.y-1)*log(mub.star[i]) -(beta.b+tau.vec)*mub.star[i] +
             (gamma.a+s.n-cum.y-1)*log(mua.star[i]) -(delta.a+n-tau.vec)*mua.star[i]
  prob.vector <- exp(temp)
  #prob.vector <- mua.star[i]^(alpha.b+cum.y-1)*exp(-(beta.b+tau.vec)*mub.star[i]) *
  #               mub.star[i]^(gamma.a+s.n-cum.y-1)*exp(-(delta.a+n-tau.vec)*mua.star[i])
  prob.vector <- prob.vector/sum(prob.vector)
  tau.star[i] <- sample(x=1:(n-1),size=1,prob=prob.vector)
}

tau.star <- tau.star + 1850
par(mfrow=c(2,3),oma=c(0,0,3,0))
plot(1:N,mub.star,xlab=expression(mu[b]),ylab="",main=expression(paste("Trace ",mu[b])),type="l")
plot(1:N,mua.star,xlab=expression(mu[a]),ylab="",main=expression(paste("Trace ",mu[a])),type="l")
plot(1:N,tau.star,xlab=expression(tau),ylab="",main=expression(paste("Trace ",tau)),type="l")
plot(density(mub.star),xlab=expression(mu[b]),ylab="",main=expression(mu[b]),type="l")
plot(density(mua.star),xlab=expression(mu[a]),ylab="",main=expression(mu[a]),type="l")
plot(density(tau.star),xlab=expression(tau),ylab="",main=expression(tau),type="l")
par(mfrow=c(1,1))

#-- summary statistics
burnin <- 10000
summary(mub.star[-c(1:burnin)])
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 2.008   2.917   3.110   3.118   3.307   4.449

summary(mua.star[-c(1:burnin)])
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.4906  0.8411  0.9177  0.9224  0.9984  1.4646

summary(tau.star[-c(1:burnin)])
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 1880    1889    1890    1890    1891    1901

theta.star <- mcmc(data=cbind(mub.star,mua.star,tau.star),start=Burnin,end=N)
MonteCarlo.se <- batchSE(x=theta.star,batchSize=round(N/30))
print(MonteCarlo.se)
#   mub.star     mua.star     tau.star
# 0.0017489203 0.0006727873 0.0126560505

round(effectiveSize(theta.star))
# mub.star mua.star tau.star
# 41322    42955    38232
```