Gibbs Samplers

Jonathan Navarrete

July 2, 2017

Introduction

Now that we've familiarized ourselves with MCMC and the Metropolis-Hastings algorithms, we begin to analyze a now-common MCMC algorithm called Gibbs sampler. The Gibbs sampler is in fact a special case of the Metropolis-Hastings algorithm for high dimensional target distributions.

We introduce the Gibbs sampler with a two-stage example. The **two-state Gibbs sampler algorithm** as described Robert & Casella goes as follows

`Take` $X_0 = x_0$

For $t = 1, 2, ...,$ `generate`

1. $y_i \sim f_{Y|X}(\cdot|x_{t-1})$
2. $X_t \sim f_{X|Y}(\cdot|y_t)$

The *two-stage* Gibbs sampler creates a Markov chain from a joint distribution in the following way. If two random variables X and Y have joint density $f(x, y)$, with corresponding conditional densities $f_{Y|X}$ and $f_{X|Y}$, the two stage Gibbs sampler generates a Markov chain (X_t, y_i) by generating y_i from conditional density $f_{Y|X}$ and then generating X_t from conditional density $f_{X|Y}$.

We illustrate the implementation of the Gibbs sampler with a simple example. Consider a bivariate Normal distribution where

$$X, Y \sim N_2\left(\begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}, \begin{pmatrix}\sigma_X^2 & \sigma_{XY}^2 \\ \sigma_{YX}^2 & \sigma_Y^2\end{pmatrix}\right)$$

The marginal distributions of X and Y are $N(\mu_X, \sigma_X)$ and $N(\mu_X, \sigma_X)$. The conditional distributions of Y and X are

$$Y|X = x \sim N\left(\mu_Y + \frac{\rho\sigma_Y}{\sigma_X}(x - \mu_X), (1 - \rho^2)\sigma_Y^2\right)$$

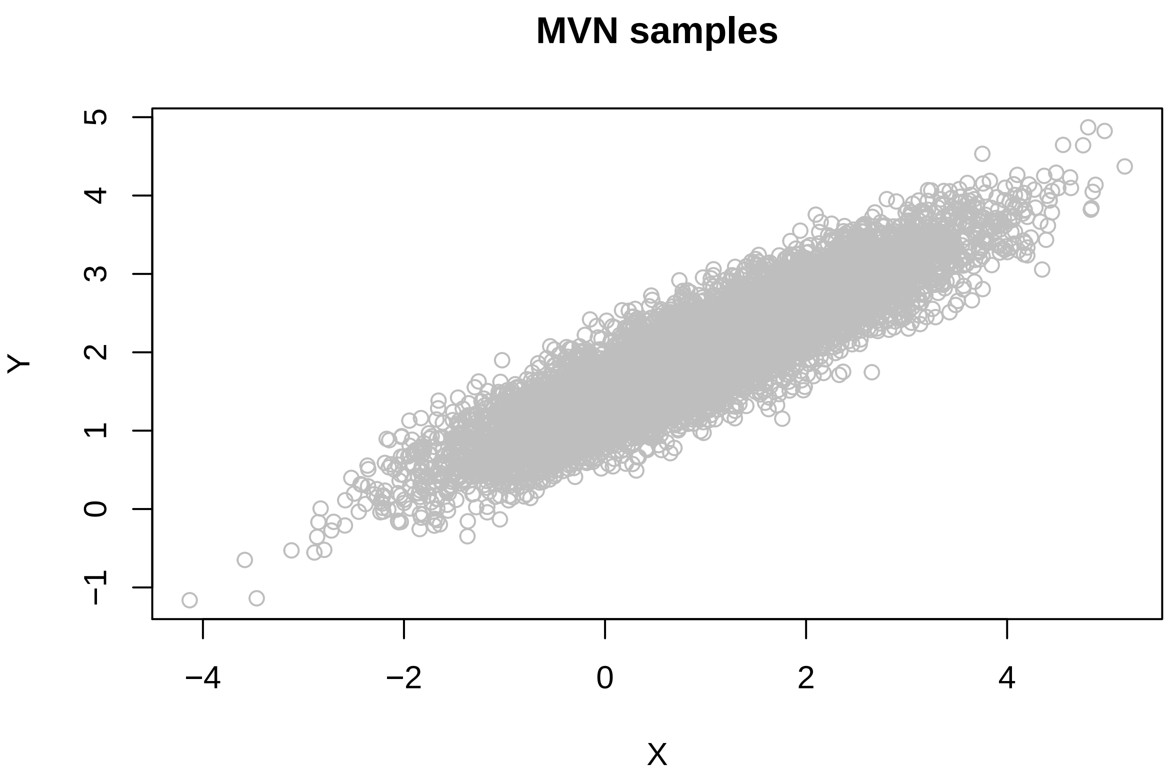
and

$$X|Y = y \sim N\left(\mu_X + \frac{\rho\sigma_X}{\sigma_Y}(y - \mu_Y), (1 - \rho^2)\sigma_X^2\right)$$

ρ is the correlation between X and Y, and $(1 - \rho^2)\sigma_X^2$ is the variance.

```
N = 10000

rho = 0.9
mu_x = 1
mu_y = 2
sd_x = 1.2
sd_y = 0.75

s1 = sqrt(1 - rho^2) * sd_x
s2 = sqrt(1 - rho^2) * sd_y

MVN = matrix(data = NA, nrow = N, ncol = 2,
          dimnames = list(NULL, c("X", "Y")))
```

```r
MVN[1, ] = c(mu_x, mu_y)
Y = MVN[1, 2] ## get Y vals
for(i in 1:(N-1)){
  mx = mu_x + rho * (Y - mu_y) * sd_x/sd_y
  X = rnorm(n = 1, mx, s1)
  MVN[i+1, 1] = X
  my = mu_y + rho * (X - mu_x) * sd_y/sd_x
  Y = rnorm(n = 1, mean = my, sd = s2)
  MVN[i+1, 2] = Y
}


plot(MVN, type = "p", col = 8,
     main = "MVN samples")
```




```r
## means
colMeans(MVN)
```

```
##         X         Y
## 0.9569281 1.9745901
```

```r
## correlation
cor(MVN)
```

```
##           X         Y
## X 1.0000000 0.9011064
## Y 0.9011064 1.0000000
```

Beta-Binomial revisited

In the introduction to these notes, we saw a Bayesian example of the Beta-Binomial distribution. From Casella's paper *Explaining the Gibbs Sampler*, we revisit this example.

$$X|\theta \sim Bin(n, \theta), \text{ and } \theta \sim Beta(a, b)$$

have joint density

$$f(x, \theta) = \binom{n}{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{x+a-1}(1-\theta)^{n-x+b-1}$$

is the a $Beta(x + a, n - x + b)$ distribution.

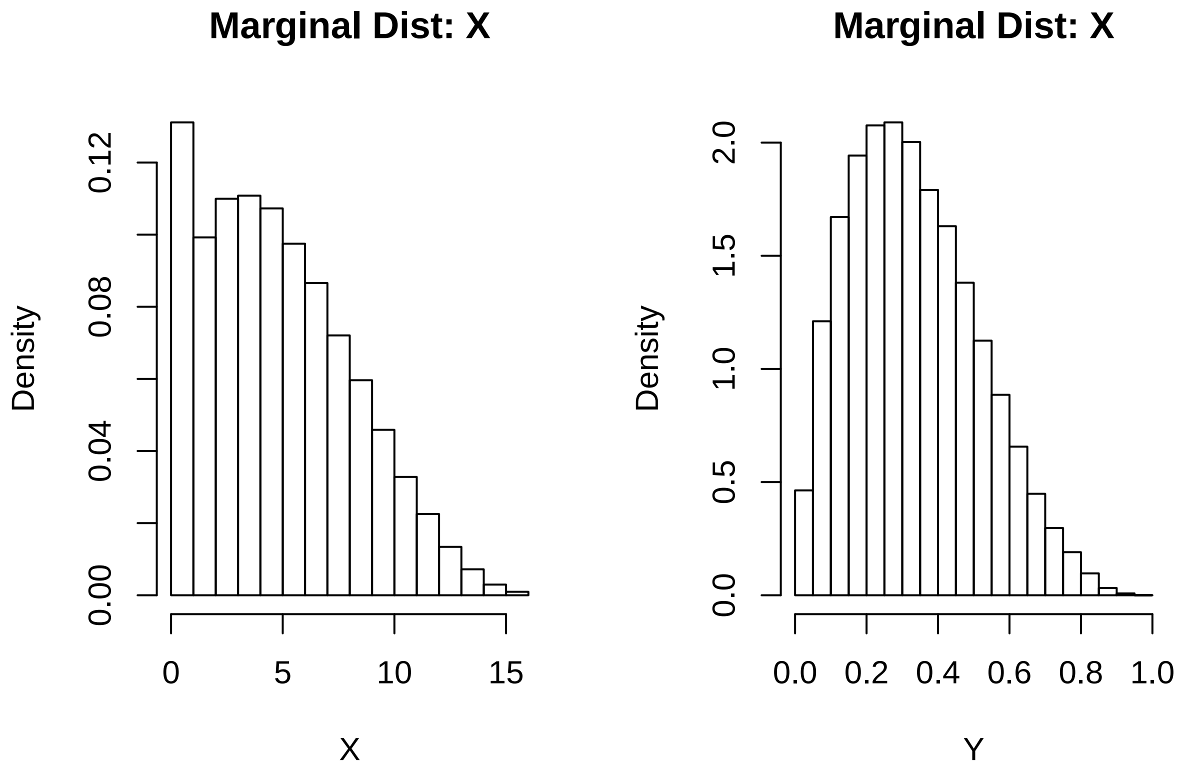
Suppose we are interested in calculating some characteristics of the marginal distributions of $X|\theta$ and $\theta|a, b, x, n$.

$$f(x|\theta) \text{ is } Bin(n, \theta) f(\theta|x) \text{ is } Beta(x + a, n - x + b)$$

Therefore, we follow an iterative algorithm of

$$X_i \sim f(x|\theta) Y_{i+1} \sim f(y|X_i = x_i)$$

```
N = 10^5
n = 16
a = 2
b = 4
X = numeric(N)
Y = numeric(N)

## initial values
X[1] = 0.2
Y[1] = 0.34 ## theta values

for(i in 1:N){
  X[i+1] = rbinom(n = 1, size = n, prob = Y[i])
  Y[i+1] = rbeta(1, a + X[i+1], n - X[i+1]+b)
}

par(mfrow = c(1,2))
hist(X, main = "Marginal Dist: X", probability = TRUE)
hist(Y, main = "Marginal Dist: X", probability = TRUE)
```


Marginal Dist: X
Density
0.00
0.04
0.08
0.12
0
5
10
15
X
Marginal Dist: X
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
Y

Poisson-Gamma Model

Consider a random sample $\mathbf{y} = (y_1, ..., y_n)^T$ where $y_i|\theta \sim Poisson(\theta)$ where $\theta \sim Gamma(\alpha, \beta)$ for $i \in 1, ..., k$. Assume t and α are known constants, but that β has a $Gamma(c, d)$ hyperprior with known hyperparameters c and d. This represents a three stage model where the likelihood, prior and hyperprior are defined as

$$f(y_i|\theta) = \frac{\theta^{y_i}}{y_i!}e^{-\theta}, y_i \geq 0, \theta_i > 0$$

$$g(\theta|\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}, \alpha > 0, \beta > 0$$

$$h(\beta) = \frac{d^c}{\Gamma(c)}\beta^{c-1}e^{-d\beta}, c > 0, d > 0$$

Suppose our interest is in generating samples from the marginal posterior distributions of θ, $p(\theta|\mathbf{y})$. Note that while the gamma prior is conjugate with the Poisson likelihood and the gamma hyperprior is conjugate with the gamma prior, no closed for for $p(\theta|\mathbf{y})$ exists. However, the full conditional distributions of β and the θ needed to implement the Gibbs sampler can be extracted from the full conditional $p(\theta|\mathbf{y})$

$$\begin{aligned}
p(\theta|\mathbf{y}, \beta) &\propto \prod_{i=1}^{n} f(y_i|\theta)g(\theta|\beta) \\
&\propto \theta^{n\bar{y}}e^{n\theta} \times \theta^{\alpha-1}e^{-\theta\beta} \\
&\propto \theta^{n\bar{y}+\alpha-1}e^{-\theta(n+\beta)} \\
&\propto Gamma(\theta|n\bar{y}+\alpha, n+\beta)
\end{aligned}$$

while for β we have

$$\begin{aligned}
p(\beta|\theta, \mathbf{y}) &\propto g(\theta|\beta) \times h(\beta) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta} \times \frac{d^c}{\Gamma(c)}\beta^{c-1}e^{-d\beta} \\
&\propto \beta^{\alpha+c-1}e^{-\beta(\theta+d)} \\
&\propto Gamma(\alpha+c, \theta+d)
\end{aligned}$$

The multistage Gibbs sampler

There is a natural extension from the two-stage Gibs sampler to the general multistage Gibbs sampler. Suppose that, for some $p > 1$, the random variable $\mathbf{X} \in X$ can be written as $\mathbf{X} = (X_1, ..., X_p)^T$ where the X_i's are either unidimensional or multidimensional components. Suppose that we can simulate from the corresponding conditional densities, $f_1, f_2, ..., f_p$ that is, we can simulate

$$X_i|x_1, ..., x_{i-1}, x_{i+1}, ..., x_p \sim f(x_i|x_1, ..., x_{i-1}, x_{i+1}, ..., x_p)$$

for i in 1, 2, $\ldots$, p. THe associated Gibbs sampler is giben as

At iteration $t = 1, 2, ...$, given $\mathbf{x}^{(\mathbf{t})} = (x^{(1)}, ..., x^{(p)})$, generate

1. $X_1^{(t+1)} \sim f_1(x_1|x_2^{(t)}, ..., x_p^{(t)})$
2. $X_2^{(t+1)} \sim f_2(x_2|x_1^{(t+1)}, x_3^{(t)} ..., x_p^{(t)})$ $\ldots$

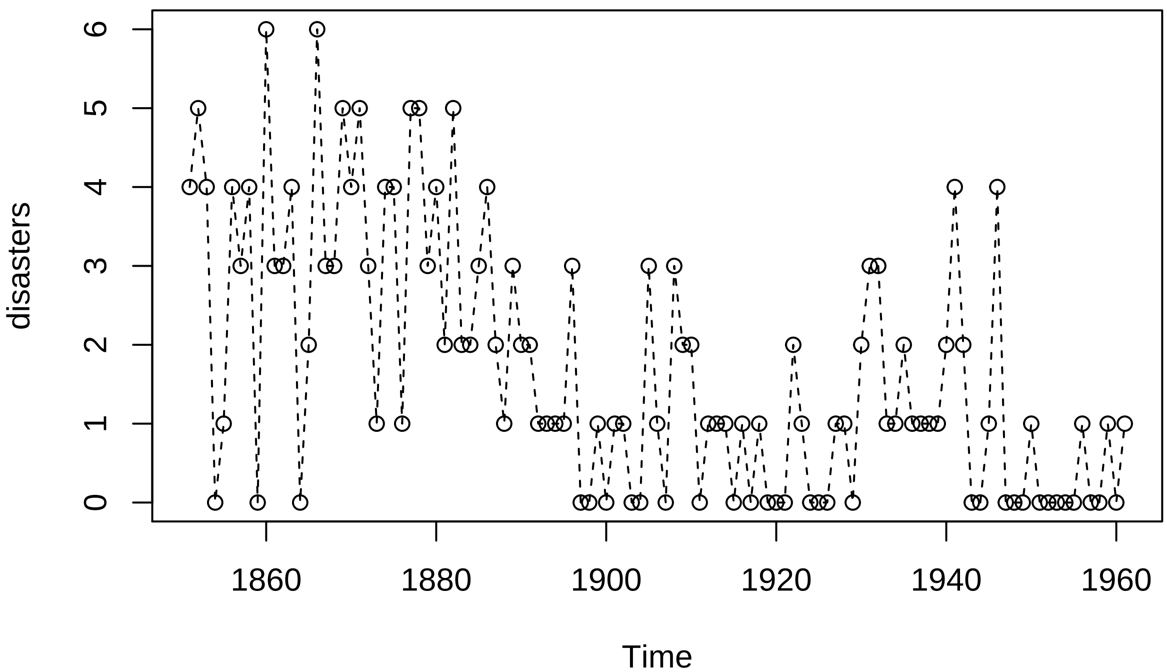
p. $X_p^{(t+1)} \sim f_p(x_p|x_1^{(t+1)}, ..., x_{p-1}^{(t+1)})$.

The densities $f_1, ..., f_p$ are called the *full conditionals*, and a particular feature of teh Gibbs sampler is that these are the only densities used for simulation. Thus, even in a high-dimensional problem, all of the simulations *may* be univariate, which can be a huge advantage!

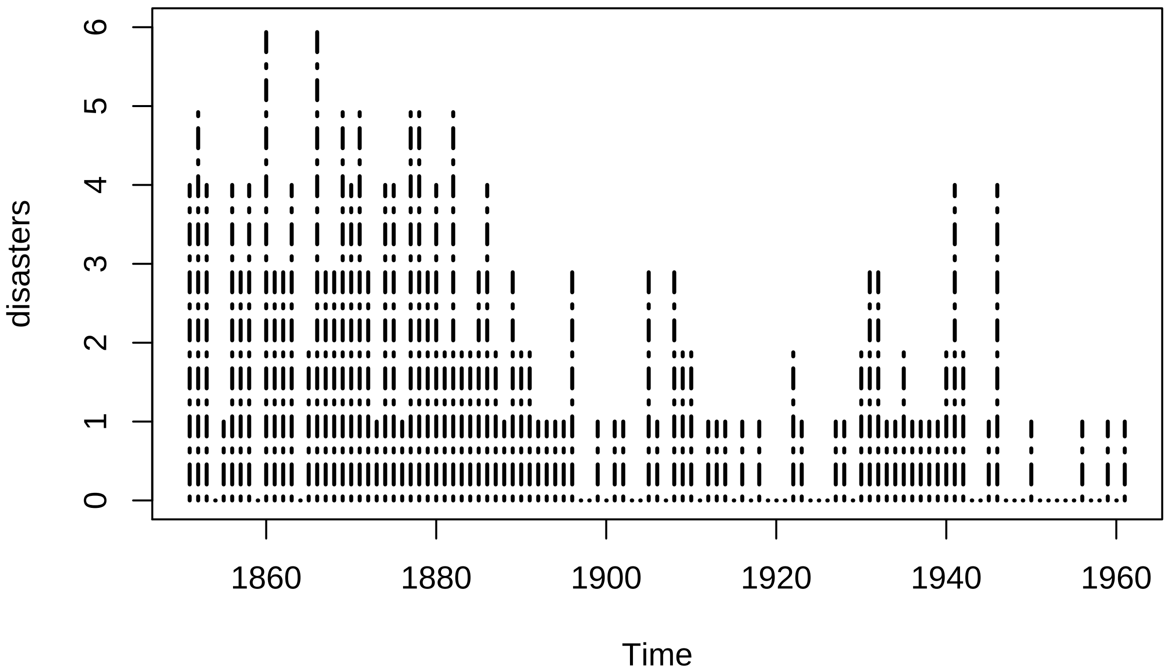
Bayesian Change-Point Analysis

Let's try to model a more interesting example, a time series of recorded coal mining disasters in the UK from 1851 to 1962, for $n = 111$ years of data. Occurrences of disasters in the time series is thought to be derived from a Poisson process with a large rate parameter in the early part of the time series, and from one with a smaller rate in the later part. We are interested in locating the change point in the series, which perhaps is related to changes in mining safety regulations.

```
data_vector = c(4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6,
                    3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5,
                    2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 0, 0,
                    1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
                    0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2,
                    3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4,
                    0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1)
n = length(data_vector)

disasters = ts(data_vector, freq=1, start = 1851)

plot(disasters, type = "o", lty = 2)
```




```
plot(disasters, type = "h", lty = 6, lwd = 2)
```



We are going to use Poisson random variables for this type of count data. Denoting year i's accident count by y_i,

$$y_i \sim \text{Poisson}(\Lambda)$$

The modeling problem revolves around estimating the values of the λ parameters. Looking at the time series above, it appears that the rate declines later in the time series. A changepoint model identifies a point (year) during the observation period (call it k) after which the parameter λ drops to a lower value. So we are estimating two λ parameters: one for the early period and another for the late period.

$$\Lambda = \begin{cases} \theta \text{ if } i < k \\ \lambda \text{ if } i \geq k \end{cases}$$

We need to assign prior probabilities to both θ and λ parameters. The gamma distribution not only provides a continuous density function for positive numbers, but it is also conjugate with the Poisson sampling distribution.

We will specify suitably vague hyperparameters, letting $\alpha = 1$ and allowing βs to vary.

$$\theta \sim \text{Gamma}(1, b_1)$$
$$\lambda \sim \text{Gamma}(1, b_2)$$

Since we do not have any intuition about the location of the changepoint (prior to viewing the data), we will assign a discrete uniform prior over all years 1851-1962.

$$k \sim \text{Unif(1851,1962)}$$

$$\Rightarrow p(K = k) = \frac{1}{111}$$

Implementing Gibbs sampling We are interested in estimating the joint posterior of θ, λ and k given the array of annnual disaster counts $\mathbf{y}$. This gives:

$$p(\theta, \lambda, k|\mathbf{y}) \propto p(\mathbf{y}|\theta, \lambda, k)p(\theta, \lambda, k)$$

To employ Gibbs sampling, we need to factor the joint posterior into the product of conditional expressions:

$$p(\theta, \lambda, k|\mathbf{y}) \propto p(y_{i<k}|\theta, k)p(y_{i\geq k}|\lambda, k)p(\theta)p(\lambda)p(k)$$

which we have specified as:

$$\begin{aligned}
p(\theta, \lambda, k|\mathbf{y}) &\propto \left[\prod_{t=1851}^{k} \text{Poisson}(y_i|\theta) \times \prod_{t=k+1}^{1962} \text{Poisson}(y_i|\lambda)\right] \times \text{Gamma}(\theta|\alpha, \beta) \times \text{Gamma}(\lambda|\alpha, \beta)\frac{1}{111} \\
&\propto \left[\prod_{t=1851}^{k} e^{-\theta}\theta^{y_i} \prod_{t=k+1}^{1962} e^{-\lambda}\lambda^{y_i}\right] \theta^{\alpha-1}e^{-\beta\theta}\lambda^{\alpha-1}e^{-\beta\lambda} \\
&\propto \theta^{\sum_{t=1851}^{k} y_i+\alpha-1}e^{-(\beta+k)\theta}\lambda^{\sum_{t=k+1}^{1962} y_i+\alpha-1}e^{-\beta\lambda}
\end{aligned}$$

So, the full conditionals are known, and critically for Gibbs, can easily be sampled from.

$$\theta \sim \text{Gamma}\left(\sum_{t=1851}^{k} y_i + \alpha, k + \beta\right)$$

$$\lambda \sim \text{Gamma}\left(\sum_{t=k+1}^{1962} y_i + \alpha, 1962 - k + \beta\right)$$

$$p(k|\mathbf{y}, \theta, \lambda, b_1, b_2) = \frac{L(\mathbf{y}|\theta, \lambda, b_1, b_2)}{\sum_{j=1}^{n} L(\mathbf{y}|\theta, \lambda, b_1, b_2)}$$

where the likelihood is defined as

$$L(\mathbf{y}|\theta, \lambda, b_1, b_2) = e^{(\lambda-\theta)}\left(\frac{\theta}{\lambda}\right)^{\sum_1^k y_i}$$

```r
set.seed(123)

y = data_vector
# Gibbs sampler for the coal mining change point
# initialization
n <- length(y)
 #length of the data
m <- 10^4
 #length of the chain
mu <- numeric(m)
lambda <- numeric(m)
k <- numeric(m)
L <- numeric(n)
k[1] <- sample(1:n, 1) ## tau
mu[1] <- 1
lambda[1] <- 1
a = 0.5
b1 <- 1
b2 <- 1
```

The algorithm explained by Carlin et al is simple. For $t \in \{1, 2, ..., m\}$

1. Sample $\theta_t \sim Gamma(a_1 + \sum_1^k y_i, k_{t-1} + b_{1,t-1})$
2. Sample $\lambda_t \sim Gamma(a_2 + \sum_{k+1}^n y_i, n - k_{t-1} + b_{2,t-1})$
3. Sample $b_1 \sim Gamma(a_1 + c_1, (\theta_t + d_1))$
4. Sample $b_1 \sim Gamma(a_2 + c_2, (\lambda_t + d_2))$
5. For $j \in 1, ..., n$ calculate $L(\mathbf{y}|\theta, \lambda, b_1, b_2)$, from there you'll obtain $p(k|\mathbf{y}, \theta, \lambda, b_1, b_2)$
6. Sample $k_t \sim p(k|\mathbf{y}, \theta, \lambda, b_1, b_2)$

```
# run the Gibbs sampler
for (t in 2:m){
    kt <- k[t-1]
    #generate mu
    r <- a + sum(y[1:kt])
    mu[t] <- rgamma(1, shape = r, rate = kt + b1)
    #generate lambda
    if (kt + 1 > n){
      r <- a + sum(y)
      }else{
        r <- a + sum(y[(kt+1):n])
      }
    lambda[t] <- rgamma(1, shape = r, rate = n - kt + b2)
    #generate b1 and b2
    b1 <- rgamma(1, shape = a, rate = mu[t]+1)
    b2 <- rgamma(1, shape = a, rate = lambda[t]+1)

    for (j in 1:n) {
        L[j] <- exp((lambda[t] - mu[t]) * j) *
                (mu[t] / lambda[t])^sum(y[1:j])
    }

    L <- L / sum(L)
    #generate k from discrete distribution L on 1:n
    k[t] <- sample(1:n, prob=L, size=1)
}
```

```
## set burn in
burn_in <- 1000

j <- k[burn_in:m]

print(mean(k[burn_in:m]))
```

```
## [1] 39.82791
```

```
#[1] 39.935

print(mean(lambda[burn_in:m]))
```

```
## [1] 0.939208
```

```
#[1] 0.9341033
```

```
print(mean(mu[burn_in:m]))
```

```
## [1] 3.130795
```

```
#[1] 3.108575



## Code for Figure 9.12 on page 276
# histograms from the Gibbs sampler output
par(mfrow=c(2,3))
labelk <- "changepoint"
label1 <- paste("mu", round(mean(mu[burn_in:m]), 1))
label2 <- paste("lambda", round(mean(lambda[burn_in:m]), 1))

hist(mu[burn_in:m], main="", xlab=label1,
     col="gray", border="white",
     breaks = "scott", prob=TRUE) #mu posterior

hist(lambda[burn_in:m], main="", xlab=label2,
     col="gray", border="white",
     breaks = "scott", prob=TRUE) #lambda posterior

hist(j, breaks=min(j):max(j), prob=TRUE, main="",
     col="gray", border="white",
     xlab = labelk)

par(mfcol=c(1,1), ask=FALSE) #restore display
```