Chapter 3
Bayesian Computation

Xavi Puig
Departament d'Estadística i I.O.
©Universitat Politècnica de Catalunya 2024

Chapter 3. Bayesian Computation

Remember……

Bayesian Model

$$\begin{array}{l}\tilde{y}\,|\,\theta \sim p(\tilde{y}\,|\,\theta) \\ \theta \sim \pi(\theta)\end{array} \longrightarrow p(\tilde{y}) = \int p(\tilde{y}\,|\,\theta)\pi(\theta)d\theta$$

Data → Bayes' Th.

Bayesian Model (updated)

$$\begin{array}{l}\tilde{y}\,|\,\theta \sim p(\tilde{y}\,|\,\theta) \\ \theta\,|\,y \sim \pi(\theta\,|\,y)\end{array} \longrightarrow p(\tilde{y}\,|\,y) = \int p(\tilde{y}\,|\,\theta)\pi(\theta\,|\,y)d\theta$$

Chapter 3. Bayesian Computation

In any Bayesian data analysis we must always calculate the posterior distribution:

$$\pi(\theta \mid y) = \frac{L_y(\theta)\pi(\theta)}{\int L_y(\theta)\pi(\theta)d\theta}$$

the most difficult part of the calculation is the denominator.

For example, when $\theta = (\theta_1, \theta_2, ..., \theta_p)$

$$\pi(\theta = (\theta_1, \theta_2, ..., \theta_p) \mid y) = \frac{L_y(\theta_1, ..., \theta_p)\pi(\theta_1, ..., \theta_p)}{\int_{\Omega_p} ... \int_{\Omega_1} L_y(\theta_1, ..., \theta_p)\pi(\theta_1, ..., \theta_p)d\theta_1 ... d\theta_p}$$

Chapter 3. Bayesian Computation

Example: Systolic blood pressure

Exercise 4.2 Systolic blood pressure: Estimating the mean of a normal location model. It has been established that the standard deviation of the systolic blood pressure everywhere in the world is 13, but its mean varies slightly from country to country around an overall worldwide mean equal to 125. The department of Health of Andorra wants to:

a) compute a point estimate for the mean of the systolic blood pressure of the inhabitants of Andorra, μ,
b) predict the systolic blood pressure of one resident of Andorra,
c) determine whether the mean of the systolic blood pressure of Andorrans is higher than the overall world mean of the systolic blood pressure.

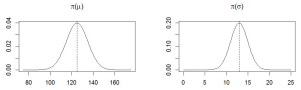
For this purpose, they obtain a sample of the systolic blood pressure of twenty Andorrans, which is: 98, 160 , 136 , 128 , 130 , 114 , 123 , 134 , 128 , 107, 123 , 125 , 129 , 132 , 154 , 115 , 126 , 132 , 136 and 130 .

Chapter 3. Bayesian Computation

Analytically:

Given the Bayesian Model:

$$\tilde{y} \mid \mu,\sigma \sim Normal(\mu,\sigma)$$
$$\mu \sim Normal(m.m = 125, s.m = 10)$$
$$\sigma \sim Normal(m.s = 13, s.s = 2)$$



and the sample $y = (y_1, y_2, \ldots, y_n)$, we calculate the posterior distribution as follows:

$$\pi(\mu,\sigma \mid y) = \frac{L_y(\mu,\sigma)\pi(\mu,\sigma)}{\iint L_y(\mu,\sigma)\pi(\mu,\sigma)\partial\mu\partial\sigma} =$$

$$= \frac{\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i-\mu)^2} \times \frac{1}{\sqrt{2\pi(s.m)^2}} e^{-\frac{1}{2(s.m)^2}(\mu-m.m)^2} \times \frac{1}{\sqrt{2\pi(s.s)^2}} e^{-\frac{1}{2(s.s)^2}(\sigma-m.s)^2}}{\iint \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i-\mu)^2} \times \frac{1}{\sqrt{2\pi(s.m)^2}} e^{-\frac{1}{2(s.m)^2}(\mu-m.m)^2} \times \frac{1}{\sqrt{2\pi(s.s)^2}} e^{-\frac{1}{2(s.s)^2}(\sigma-m.s)^2} \partial\mu\partial\sigma}$$

and the marginal posterior distributions as:

$$\pi(\mu \mid y) = \int \pi(\mu,\sigma \mid y)\partial\sigma$$
$$\pi(\sigma \mid y) = \int \pi(\mu,\sigma \mid y)\partial\mu$$

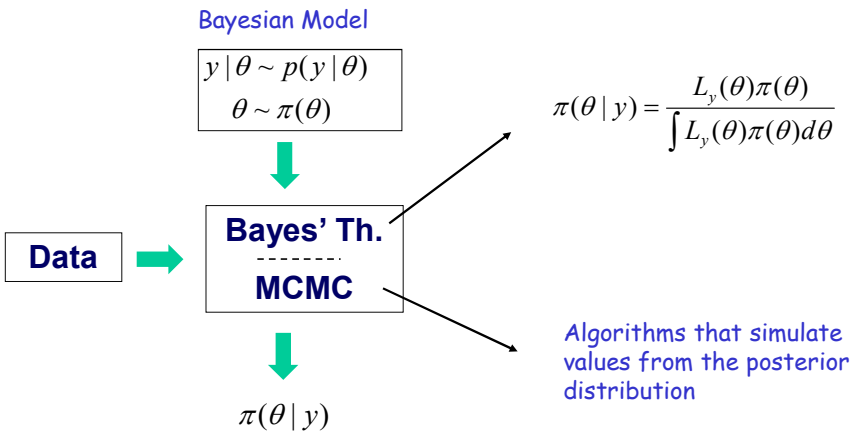
Chapter 3. Bayesian Computation

JAGS' implementation:

```
library(R2jags)

model.norm <- "
model
{
 for (i in 1:n) {
  y[i]~dnorm(mu, tau)
 }

 mu ~ dnorm(m, t)
 sigma ~ dnorm(m.s,t.s)

 tau <- 1/(sigma*sigma)

}
"

data <- list(y=c(98, 160, 136, 128, 130, 114, 123, 134, 128, 107, 123, 125, 129, 135, 154,
                 115, 126, 132, 136, 130), n=20, m=125, t=1/10^2, m.s=13, t.s=1/2^2)

parameters <- c("mu", "sigma")

model.systolyc  <-jags(data, parameters.to.save=parameters, model=textConnection(model.norm))

print(model.systolyc)
```

How can JAGS (WinBUGS or STAN) get simulations from the posterior distribution?

Chapter 3. Bayesian Computation



We call Bayesian Computation all procedures that allow us to approximate the posterior distribution.

Chapter 3. Bayesian Computation

Remember that:

- If we have simulations of the posterior distribution, we can approximate its probability distribution and any other calculation such as moments, percentiles, etc.
- The approximations will be better as the number of simulations increases.

Chapter 3. Bayesian Computation

MCMC: Markov Chain Monte Carlo

The MCMC algorithms allows from a Bayesian model and Data to simulate values of the posterior distribution

The most common MCMC algorithms are:

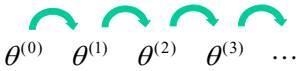
- Gibbs Sampling
- Metropolis Hasting
- Hamiltonian Montecarlo
- ....

Chapter 3. Bayesian Computation

MCMC: Markov Chain Monte Carlo

All MCMC algorithms have in common that:

- Given an initial value, $\theta^{(0)}$, simulate successive values

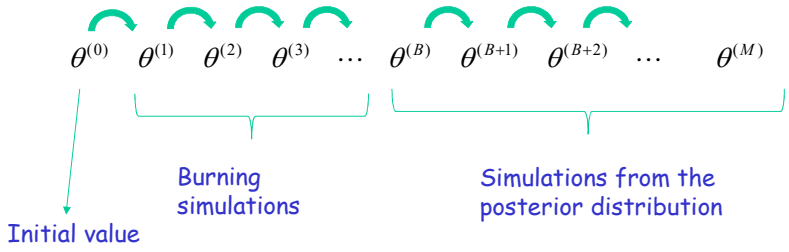


- After several simulations, the chain has converged to its stationary distribution and from now on the simulations came from the posterior distribution.

Chapter 3. Bayesian Computation

MCMC: Markov Chain Monte Carlo

MCMC algorithms simulate values of a Markov Chain where its stationary distribution is the posterior distribution.



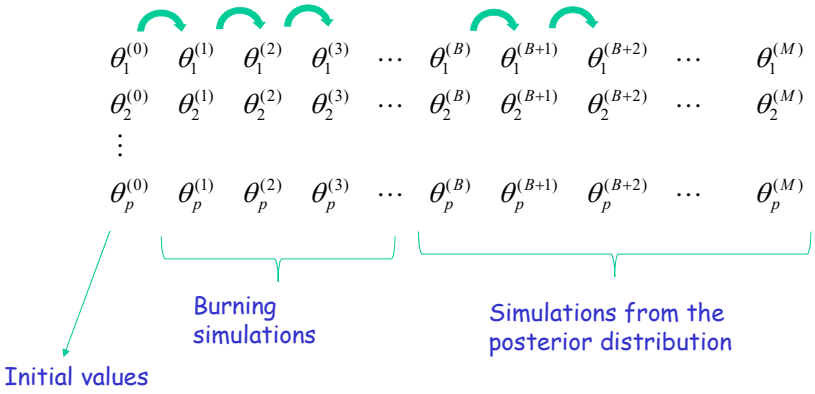
To approximate the posterior distribution we only have to use the simulations:

$$\theta^{(B)} \quad \theta^{(B+1)} \quad \theta^{(B+2)} \quad \dots \quad \theta^{(M)}$$

Chapter 3. Bayesian Computation

MCMC: Markov Chain Monte Carlo

For every parameter we have:



As users, all we need to know is B, that is, when the chains have converged to their stationary distribution and therefore they are simulating from the posterior distribution.

Chapter 3. Bayesian Computation

Checking the convergence

The convergence check is usually done graphically.

It consists of repeating the MCMC algorithm two (o more) times but starting from **different initial values**.

So for each parameter we will have 2 (or more) chains:

- Chain 1

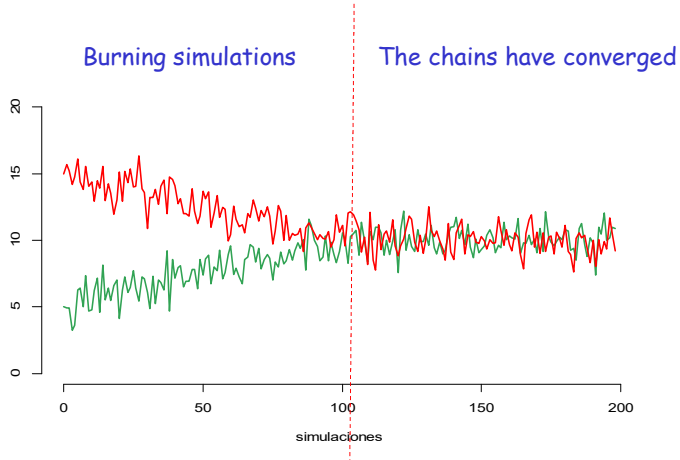
$$\theta^{(0)} \quad \theta^{(1)} \quad \theta^{(2)} \quad \theta^{(3)} \quad \dots \quad \theta^{(B)} \quad \theta^{(B+1)} \quad \theta^{(B+2)} \quad \dots \quad \theta^{(M)}$$

- Chain 2

$$\theta^{(0)} \quad \theta^{(1)} \quad \theta^{(2)} \quad \theta^{(3)} \quad \dots \quad \theta^{(B)} \quad \theta^{(B+1)} \quad \theta^{(B+2)} \quad \dots \quad \theta^{(M)}$$

Chapter 3. Bayesian Computation

Checking the convergence

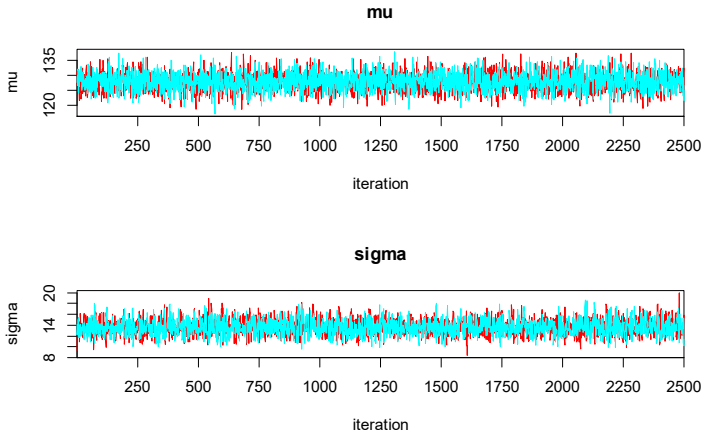


B

Chapter 3. Bayesian Computation

Example: Systolic blood pressure (continued)

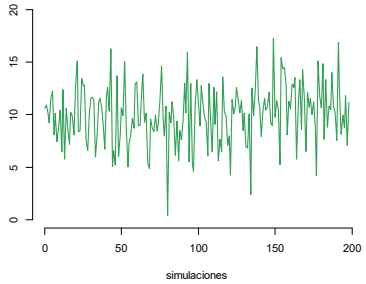
$$\tilde{y} \mid \mu, \sigma \sim Normal(\mu, \sigma)$$
$$\mu \sim Normal(m.m = 125, s.m = 10)$$
$$\sigma \sim Normal(m.s = 13, s.s = 2)$$



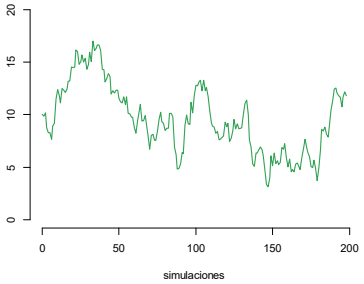
Chapter 3. Bayesian Computation

Chain autocorrelation

No autocorrelation

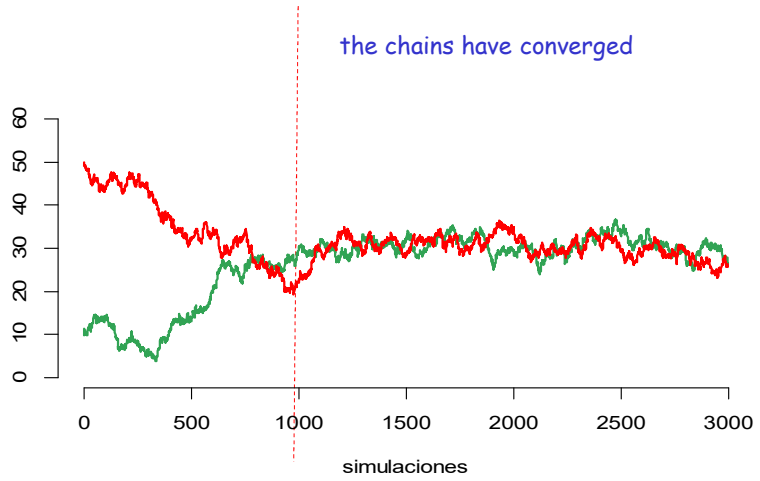


Autocorrelation



When a chain presents autocorrelation it is necessary to simulate more values to properly approximate the posterior distribution.

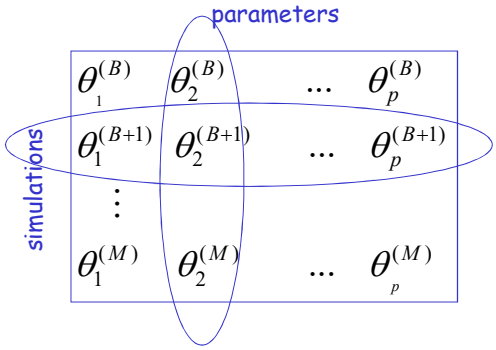
Chapter 3. Bayesian Computation



B

Chapter 3. Bayesian Computation

Simulated values



The rows are simulations from the posterior joint distribution

$$\pi(\theta_1, \theta_2, ..., \theta_p \mid y)$$

The columns are simulations from the posterior marginal distribution.

$$\pi(\theta_2 \mid y) = \int_{\Omega_p} ... \int_{\Omega_3 \Omega_1} \pi(\theta_1, ..., \theta_p \mid y) d\theta_1 d\theta_3 ... d\theta_p$$

Chapter 3. Bayesian Computation

Software:

- BUGS → The oldest, prior to Windows
- WinBUGS → The most used of all time
- OpenBUGS
- **JAGS**
- **STAN**

  → The most used today
- R libraries: https://cran.r-project.org/web/views/Bayesian.html
  INLA: https://www.r-inla.org/

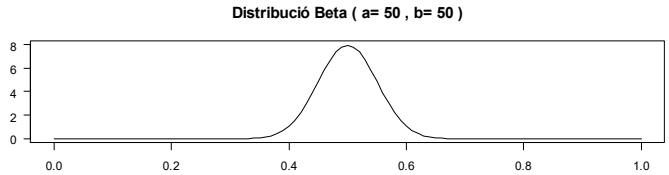
Chapter 3. Bayesian Computation

Example: Coin

The chosen model is:

$$\tilde{y} \mid \theta \sim Binomial(n = 10, \theta), \quad \theta \in (0, 1)$$

The chosen prior distribution is:

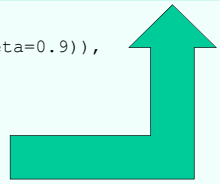


Chapter 3. Bayesian Computation

Exemple: Coin

```
library(R2jags)

model.bin <- "
 model
 {
  y ~ dbin(theta, n)
  theta ~ dbeta(a,b)
 } "

Iter <- 10000
Burn <- 500
Chain <- 2
Thin <- 3

example <-jags(data=list(n=10, y=7, a=50, b=50),
               inits=list(list(theta=0.1),list(theta=0.9)),
               parameters.to.save=c("theta"),
               model=textConnection(model.bin),
               n.iter=(Iter*Thin+Burn),
               n.burnin=Burn,
               n.thin=Thin,
               n.chains=Chain)
```

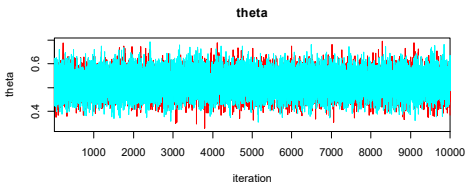
*Burn simulations are discarded, then Iter*thin values are simulated but only one of each Thin, is saved, i.e. Iter. And all this for each Chain.*



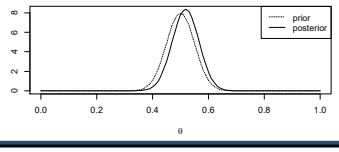
Chapter 3. Bayesian Computation

Example: Coin

```
traceplot(example, mfrow = c(1,1), varname = c("theta"))
```




```
attach.jags(example)

  plot(density(theta, adjust = 1.5), ylab="", xlab=expression(theta), main="", xlim=c(0,1))
  plot(function(x)dbeta(x,a,b),lty=3 ,add=TRUE)
   legend("topright", c("prior","posterior"), lty=c(2,1))

detach.jags()
```



Chapter 3. Bayesian Computation

Summarizing:

- The MCMC algorithms are tools that allow us to simulate from the posterior distribution starting from the Bayesian model and the data.
- As a user, when you use the MCMC algorithms to get simulations from the posterior distribution, you must check the convergence of the chains before making any inference.