MRC Biostatistics Unit

UNIVERSITY OF CAMBRIDGE

# Bayesian Methods for Clinical Trials

*Lecture 4: Implementation of Bayesian designs in JAGS*

Libby Daniells & Pavel Mozgunov & Thomas Jaki

MRC Biostatistics Unit

February 24, 2025

# Multiple parameters

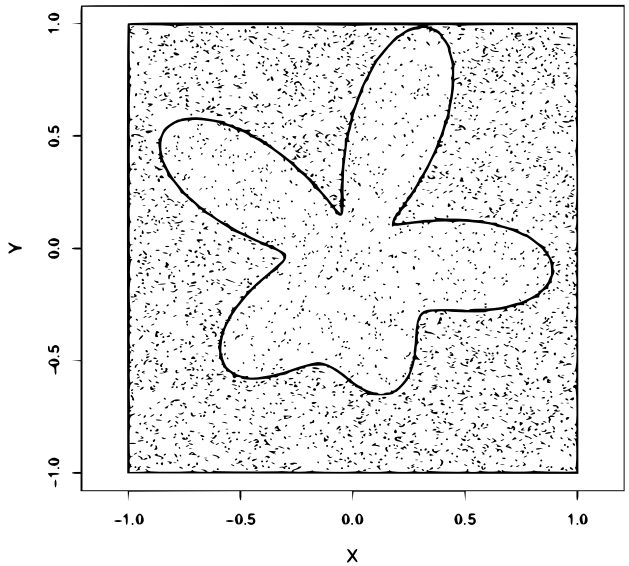
In many multiple parameter scenarios, a closed form solution is not possible for the posterior distribution.

However, we are often able to generate samples from this distribution and thereby estimate useful properties about the joint posterior

This type of calculation is called a Monte Carlo estimate.

# Monte Carlo estimate

Estimate the expectation of $(X, Y) \sim \mathcal{U}(\mathcal{A})$ for region $\mathcal{A}$ where $\mathcal{U}(\mathcal{A})$ is a bivariate uniform distribution over region $\mathcal{A}$.




Sample uniformly over $[-1, 1] \times [-1, 1]$. Keep sample of $(x_i, y_i) \in \mathcal{A}$, else discard. Repeat until we have $n$ samples

$$\mathbb{E}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i = 0.000 \qquad \mathbb{E}(Y) = \frac{1}{n}\sum_{i=1}^{n} y_i = 0.078$$

# Markov Chains

How do we generate samples from the joint posterior?

If the conditional posterior distributions are known, then we can sample from the joint posterior using a **Markov chain**.

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event

*i.e. the process for which predictions can be made regarding future outcomes based solely on its present state*

The process of using a Markov chain to draw samples from the posterior distribution is called *Markov chain Monte Carlo* (MCMC) sampling.

# Aim of MCMC

Construct a Markov chain whose stationary distribution is the *posterior* distribution we are interested in.

Therefore a sample from the Markov chain will form a sample from the posterior distribution of interest.

It is important to check that the properties of the Markov chain that we construct.

# Gibbs Sampling: Introduction

The Gibbs sampler obtains the samples of interest by successively and repeatedly simulating from the conditional distributions of each component given the other components.

1. Initialise with $\theta = (\theta_1^{(0)}, \ldots, \theta_d^{(0)})$.

2. For $i = 1, 2, \ldots, I$,
   - 2.1 Simulate $\theta_1^{(i)}$ from the conditional $\theta_1 | (\theta_2^{(i-1)}, \ldots, \theta_d^{(i-1)})$
   - 2.2 Simulate $\theta_2^{(i)}$ from the conditional $\theta_2 | (\theta_1^{(i)}, \theta_3^{(i-1)}, \ldots, \theta_d^{(i-1)})$
   - 2.3 ...
   - 2.4 Simulate $\theta_d^{(i)}$ from the conditional $\theta_d | (\theta_1^{(i)}, \ldots, \theta_{d-1}^{(i)})$

3. Discard the first $k$ iterations and estimate summary statistics of the posterior distribution using $(\theta_1^{(k+1)}, \ldots, \theta_d^{(k+1)}), \ldots, (\theta_1^{(I)}, \ldots, \theta_d^{(I)})$.

# Normal example

For the simple normal example:

1. Initialise with $\theta = (\mu_0, \tau_0)$.

   Any values of $\mu_0$ and $\tau_0 > 0$ will be OK.

2. For $k = 1, 2, \ldots, n$:

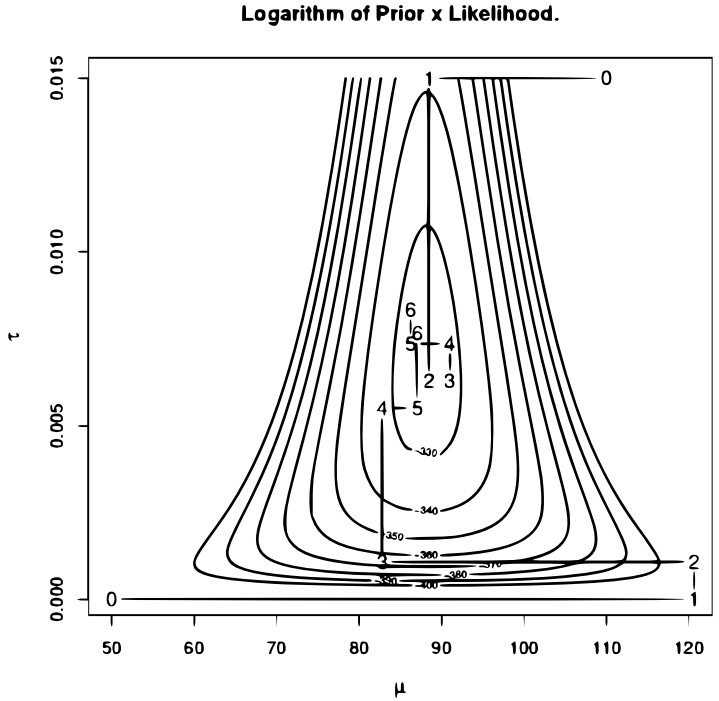
   a) Simulate $\mu_k | \tau_{k-1} = \tau, \mathbf{x} \sim N\left(\frac{\tau \sum x_i + \alpha\beta}{n\tau + \beta}, \frac{1}{n\tau + \beta}\right)$

   b) Simulate $\tau_k | \mu_k = \mu, \mathbf{x} \sim Gamma\left(\gamma + \frac{n}{2}, \delta + \frac{1}{2}\sum_i (x_i - \mu)^2\right)$.

# Gibbs sampler: Normal Example

Set initial values of $(\mu_0, \tau_0)$. Iterate $i = 1, 2, \ldots$

- Sample $\mu_i$ from conditional posterior $\mu|\boldsymbol{x}, \tau = \tau_{i-1}$
- Sample $\tau_i$ from conditional posterior $\tau|\boldsymbol{x}, \mu = \mu_i$




The Markov chain converges to the stationary distribution constructed to be the posterior. Must discard the initial set of samples.

# Gibbs sampler algorithm

Under mild regularity conditions, the convergence of the Markov chain to the stationary (posterior) distribution $P(\theta_1, \theta_2, \ldots, \theta_d)$ is guaranteed.

Therefore after a *burn-in* period, the observations

$$\theta^{(k+1)}, \theta^{(k+2)}, \ldots, \theta^{(l)}$$

can be regarded as realisations from this distribution.

# What is the burn-in?

We have to specify initial values for $\theta$. If $\theta^{(0)}$ was drawn from the stationary distribution of the Markov chain, then

$$\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(l)}$$

would be realisations from the stationary distribution.

However, we don't know what the (stationary) distribution is. (If we did, why use MCMC?)

Thus typically $\theta^{(0)}$ will not be drawn from the stationary distribution, and if $\theta^{(0)} \overset{D}{\neq} \pi$, then for all $i = 1, \ldots, k$,

$$\theta^{(i)} \overset{D}{\neq} \pi.$$

# Burn-in

However, as $k$ increases $\theta^{(k)}$ increasingly 'forgets', the initial value $\theta^{(0)}$, and consequently, for large $k$, $\theta^{(k)}$ is approximately from the stationary distribution.

That is,

$$\theta^{(k)} \overset{D}{\approx} \pi.$$

Hence,

$$\theta^{(k+1)}, \theta^{(k+2)}, \ldots, \theta^{(l)}$$

are approximately from the stationary distribution

# JAGS

- To implement the inference for more involving distributions and various type of data, we shall be using JAGS.
- JAGS stands for Just Another Gibbs Sampling.

# JAGS

- JAGS can be used from R.
- JAGS is free for download at:
  `https://jags.software.informer.com/4.0/`
- There are many examples and resources online.

# JAGS

- JAGS has a very R like language at its base.
- Six stages to running MCMC with JAGS:
  1. Specify the model - entails specification of the conditional distribution of the data.
  2. Specify the prior distributions.
  3. Feed in data.
  4. Give some initial values for the parameters.
  5. Tell JAGS what to “monitor”.
  6. Run the model.

# Illustration

We start with the illustration of `rjags` package in `R`.

For illustration purposes, we will use a normal example for which we do now the analytical formula for the posterior distribution.

Imagine that we have a sample of 20 observations $Y_1, \ldots, Y_{20}$ having

$$Y_i \sim \mathcal{N}(\mu, 1)$$

with the known variance $\sigma^2 = 1$.

We impose a vague prior $\mu \sim \mathcal{N}(0, \tau = 0.01)$ where the precision $\tau = 0.01$.

# Illustration

We generate data using the true value of $\mu = 1$. Given these realisations, we would like to obtain the posterior distribution.

## R code

```
set.seed(100)
data<-list(Y=rnorm(20,mean=1,sd=1),N=20)
```

Knowing the analytical solutions we can obtain the parameters of the normal posterior

## R code

```
mean(data$Y)
[1] 1.107867 # posterior mean
sqrt(1/(data$N+0.01))
[1] 0.2235509 # posterior standard deviation
```

# Illustration

Alternatively, we can use JAGS to derive the posterior distribution, or, in fact, the samples from the posterior distribution.

The models can be specified as:

R code

```
install.packages("rjags")
library(rjags)

model1.string <-"model {
  for (j in 1:N) {
    Y[j] ~ dnorm(mu, 1) # Responses Y are Normal
  }
  ## Prior
  mu ~ dnorm(0, 0.01)
}"
```

# Illustration

## R code

```
inits<-list(mu=0)
iter<-10000

model1.spec<-textConnection(model1.string)
jags <- jags.model(model1.spec, data = data,
inits=inits, n.chains = 1,quiet=TRUE,n.adapt=iter)

update(jags,iter,progress.bar="none")

# Storing Samples of the Posterior Distribution
sampling<-jags.samples(jags,c('mu'),
iter,progress.bar="none")
samples<-sampling$mu[1,,]
```

# Illustration

## R code

```
inits<-list(mu=0)
iter<-10000

model1.spec<-textConnection(model1.string)
jags <- jags.model(model1.spec, data = data,
inits=inits, n.chains = 1,quiet=TRUE,n.adapt=iter)

update(jags,iter,progress.bar="none")

# Storing Samples of the Posterior Distribution
sampling<-jags.samples(jags,c('mu'),
iter,progress.bar="none")
samples<-sampling$mu[1,,]
```

# Illustration

## R code

```
> mean(samples)
[1] 1.106733
> sd(samples)
[1] 0.2204946

###

mean(data$Y)
[1] 1.107867 # posterior mean
sqrt(1/(data$N+0.01))
[1] 0.2235509 # posterior standard deviation
```

This is in line with the analytical solution obtained above.

# Illustration

Using the samples, we can get any summary characteristics that we would like to know. For example, what is the probability that $\mu \geq 1$.

Using the analytical solution, we can get that

R code

```
> pnorm(1,mean=1.107867,sd=0.2235509,lower.tail=F)
[1] 0.6852805
```

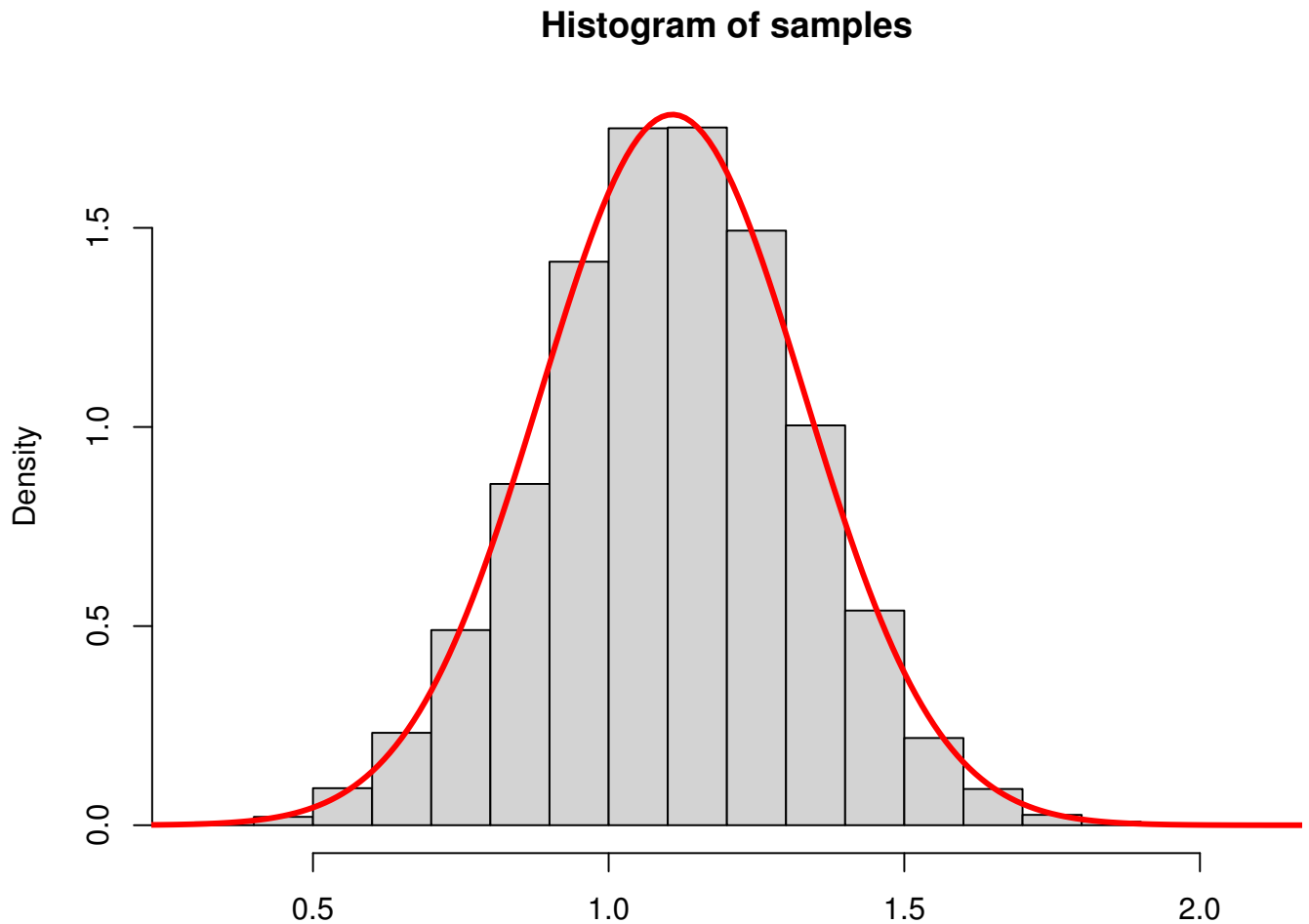
Or using the posterior samples

R code

```
> mean(samples>1)
[1] 0.6885
```

# Distribution of Samples

We can compare the histogram of samples and the posterior density function using the analytical formula.



# Further extension

Imagine now, that there is a prior knowledge that the mean response belong to the interval $(0, 1.25)$ with any of these values being equally possible.

So, $\mu \sim \mathcal{U}(0, 1.25)$ is our prior. It is not conjugate, and, therefore, there is no analytical solution for the posterior.

However, we can still use JAGS with very little change to the code.

# Updated Model

Our updated model will be

## R code

```
install.packages("rjags")
library(rjags)

model1.string <-"model {
  for (j in 1:N) {
    Y[j] ~ dnorm(mu, 1) # Responses Y are Normal
  }
  ## Prior
  mu ~ dunif(0, 1.25)
}"
```

# Updated summaries

The rest of the code remains the same. The summary of the sample are

R code

```
> mean(samples)
[1] 1.013172
> sd(samples)
[1] 0.1610121
```

# Updated histogram