7.4 MCMC in BUGS

Zongyi Liu

2023-06-15

7.4 MCMC in BUGS

7.4.1 Parameter Estimation with BUGS

MCMC sampling of a posterior distribution is simple in BUGS. We merely need to specify the prior, the likelihood, and the observed data.

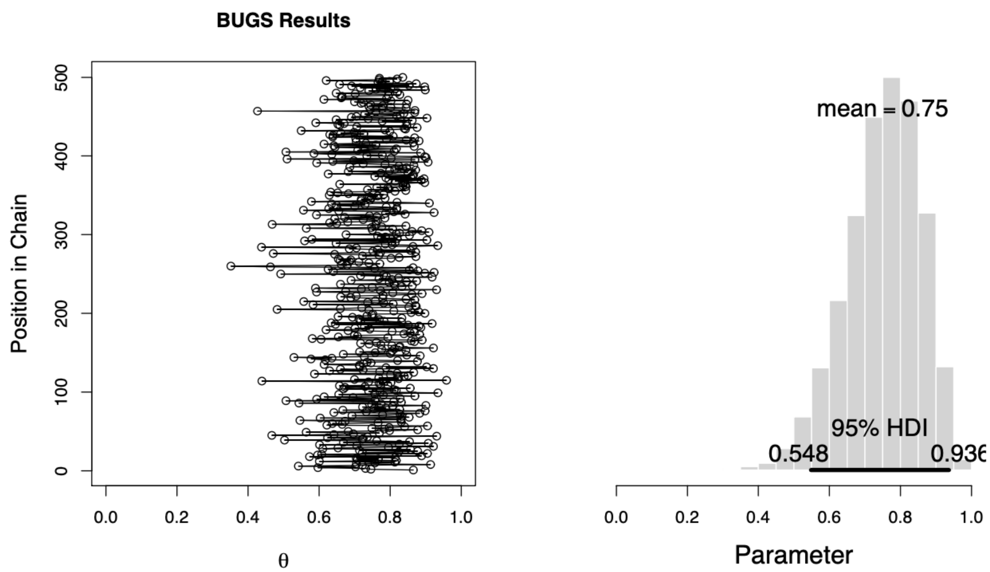
Here we have a beta(1,1) prior distribution, a Bernoulli likelihood function, and data consisting of 14 flips with 11 heads. The prior distribution and likelihood function can be specified as

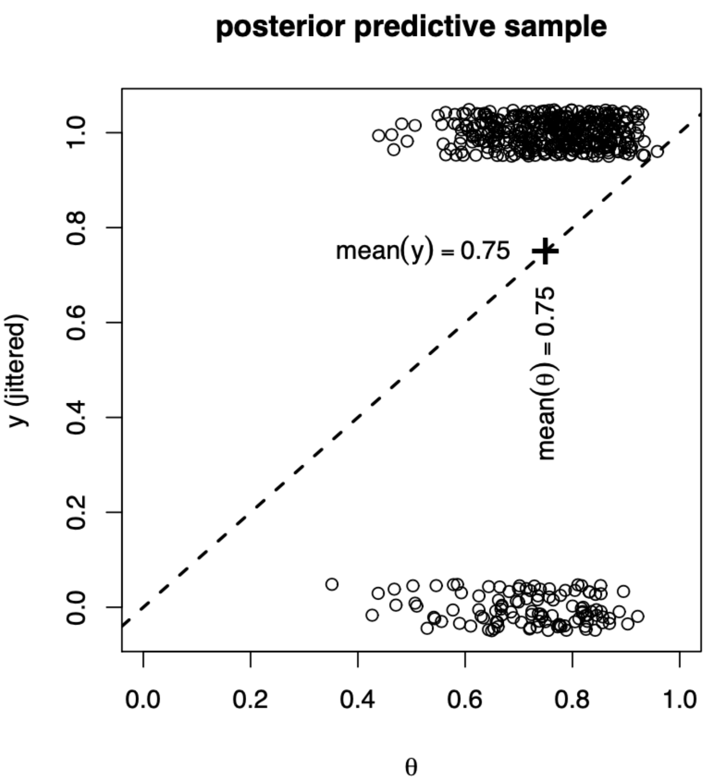
```
model {
    # Likelihood:
    for ( i in 1:nFlips ) {
        y[i] ~ dbern( theta )
    }
    # Prior distribution:
    theta ~ dbeta( priorA , priorB )
    priorA <- 1
    priorB <- 1
}
```

The code in the for loop says that every flip of the coin comes from a Bernoulli distribution with parameter value theta, and the code at the end says that the value of `theta` comes from a prior distribution that is `beta` with shape parameters `priorA` and `priorB`.

7.4.2 BUGS for prediction

The goal for prediction is determining the probability of subsequent data values. As described before, when the likelihood function is Bernoulli, then mathematical derivation tells us that $p(y = 1|D)$ is the mean of the posterior distribution of θ. We can forgo the mathematical analysis when we use MCMC sampling.





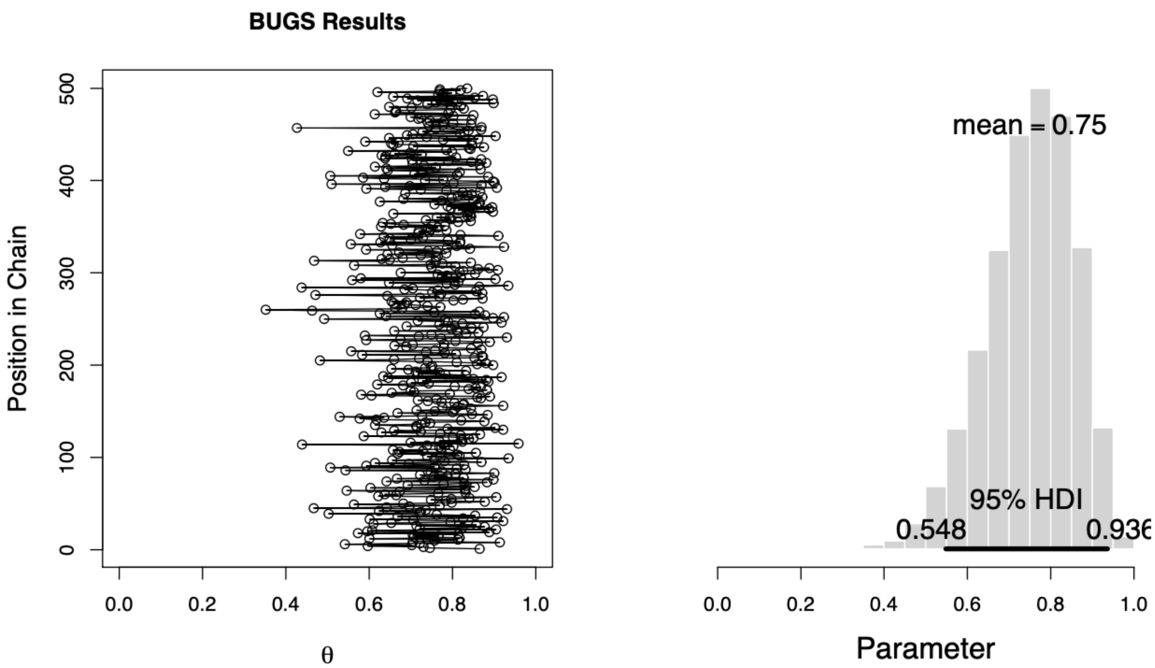
7.4.3 BUGS for Model Comparison

BUGS provides no short-cut for estimate $p(D)$ relative to the homegrown Metropolis algorithm.

If we use the method of section 7.3.3, when the likelihood function is Bernoulli, the mathematical derivation tells us that $p(y = 1|D)$ is the mean of the posterior distribution of θ. We can forgo the mathematical analysis when we use MCMC sampling.

Instead, we can directly generate simulated values of y from the posterior sampled values of θ, and then examine the distribution of y.

To generate random y values based on the posterior sampled values of θ, we add a few lines of R code at the end of the program. Namely, we can flip the coin, and the command to flip the coin in R is `sample()`. The arguments of `sample()` are `x=c(0,1)`, which indicates the values to be sampled from, `prob=c(1-pHead,pHead)`, which indicates the probability of each value, and `size=1`, which indicates to flip the coin just once. We can have the graph plotted as below



The figure above shows the results of the posterior predicted values of y. The actual values of y are 1 or θ, but they have been jittered in the graph to make them more visible.

7.4.3 BUGS for Model Comparison

BUGS provides no short cut for estimating $p(D)$ relative to the homegrown Metropolis algorithm. If we use the method of Section 7.3.3 to estimate $p(D)$, then we still have to invent a posterior-mimicking distribution and compute the likelihood times prior at every point of the MCMC chain. The only advantage of using BUGS in this case is that the sampling algorithm for generating the chain is done for us.