Using JAGS for MCMC sampling

Chapter 3.3: Introduction to JAGS

In this example, we use JAGS to conduct simple linear regression. Before executing this code, but sure to install *JAGS* and the *R* package *rjags*.

The response is the mass of a T. Rex and the covariate is the age. The model is

```
mass_i \sim \text{Normal}(\beta_1 + \beta_2 age_i, \sigma^2).
```

The priors are $\beta_1, \beta_2 \sim \text{Normal}(0, 1000)$ and $\sigma^2 \sim \text{InvGamma}(0.1, 0.1)$.

Load T-Rex data

(1) Define the model as a string

```
model_string <- textConnection("model{

# Likelihood (dnorm uses a precision, not variance)
for(i in 1:n){
    mass[i] ~ dnorm(beta1 + beta2*age[i],tau) #tau = 1/sigma^2
}

# Priors
tau ~ dgamma(0.1, 0.1)
sigma <- 1/sqrt(tau)
beta1 ~ dnorm(0, 0.001)
beta2 ~ dnorm(0, 0.001)</pre>
```

(2) Load the data and compile the MCMC code

```
inits <- list(beta1=rnorm(1),beta2=rnorm(1),tau=10)
model <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)</pre>
```

(3) Burn-in for 10000 samples

```
update(model, 10000, progress.bar="none")
```

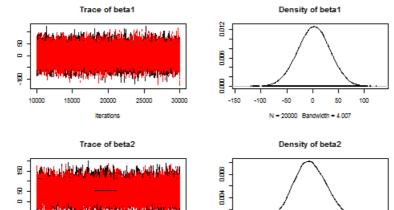
(4) Generate 20000 post-burn-in samples and retain the parameters named in params

(5) Summarize the output

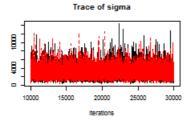
```
summary(samples)
```

```
##
## Iterations = 10001:30000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##
     plus standard error of the mean:
##
##
                      SD Naive SE Time-series SE
## beta1
           2.477
                   31.47
                          0.1573
                                          0.1573
          51.979
                   39.04
                          0.1952
                                          0.3849
## beta2
                                         10.0124
## sigma 2809.370 1169.48
                          5.8474
## 2. Quantiles for each variable:
##
                                          97.5%
                    25%
##
           2.5%
                             50%
                                     75%
## beta1 -59.55 -18.58
                           2.723
                                   23.68
                                           63.58
## beta2 -20.88
                 25.25
                          50.475
                                   77.65 132.84
## sigma 1095.13 2015.22 2620.528 3384.47 5647.02
```

plot(samples)

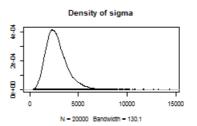


-100 -50



20000

15000



N = 20000 Bandwidth = 4.97

Loading [MathJax]/jax/output/HTML-CSS/jax.js