OpenBugs

Michael Andreae

December 11, 2015

We are trying to repeat the analysis by Dr. Johnson's for the ICBG evidency synthesis for ACE 151

OpenBugs Example

We followed the Tutorial

```
model <- function() {
    # Prior
    p ~ dbeta(1, 1)

    # Likelihood
    y ~ dbin(p, N)
}

# To transfer the model to OpenBUGS, we load the R2OpenBUGS extension
# and write the model to a temporary location using the method
# write.model. We denote the model file location by model.file.

library(R2OpenBUGS)
model.file <- file.path(tempdir(),
                        "model.txt")
write.model(model, model.file)

# Problem
# The data set survey contains sample smoker statistics among
# university students. Denote the proportion of smokers in the general
# student population by p. With uniform prior, find the mean and
# standard deviation of the posterior of p using OpenBUGS.

# Then we have to decide data parameters of the BUGS model. We find
# that there are 236 students in the survey, and 47 of them smoke,
# which we denote by N and y respectively.

library(MASS)
tbl <- table(survey$Smoke)
N <- as.numeric(sum(tbl)); N
```

```
## [1] 236
```

```
y <- N - as.numeric(tbl["Never"]); y
```

```
## [1] 47
```

```
# We then identify data variables in a list called data.

data <- list("N", "y")

# And we identify the variable p to be monitored in a vector called
# params.

params <- c("p")

# Lastly, we need to select some initial parameters for the simulation.
# A rule of thumb is to choose values as close to the expected result
# as possible. In this case, we initialize p to be 0.5. Notice how we
# wrap the initial values inside a list that is to be returned by a
# function.

inits <- function() { list(p=0.5) }

# Then we invoke OpenBUGS with the namesake method bugs and save the
# result in a variable out. We select 10,000 iterations per simulation
# chain.

out <- bugs(data, inits, params,  model.file, n.iter=10000)
out$summary
```

```
##               mean         sd   2.5%    25%    50%    75%    97.5%
## p        0.2014755 0.02574684 0.1531 0.1838 0.2009 0.2184  0.25450
## deviance 6.4509696 1.38373189 5.4710 5.5700 5.9070 6.7770 10.47025
##              Rhat n.eff
## p        1.001293  5100
## deviance 1.001188  6900
```

```
# Better to invoke the CODA option to get mcmc.list as output.

out <- bugs(data, inits, params, model.file, codaPkg=TRUE,
            n.iter=10000)
out.coda <- read.bugs(out)
```

```
## Abstracting deviance ... 5000 valid values
## Abstracting p ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting p ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting p ... 5000 valid values
```

```
# Analyse using shinystan

library(shinystan)
library(coda)

# After verification that the opject is now a mcmc.list
# convert to shinstan object and call shinystan

is.mcmc.list(out.coda)
```

```
## [1] TRUE
```

```
out.shiny <- as.shinystan(out.coda)
# launch_shinystan(out.shiny)
```