STAT431_Extra_Homework

Giang Le

12/13/2021

Extra Homework

Read in the data.

```
library("nlme")
Y <- reshape(Orthodont, v.names="distance", timevar="age",
idvar="Subject", drop="Sex", direction="wide")[2:5]
head(Y)
```

```
##    distance.8 distance.10 distance.12 distance.14
## 1        26.0        25.0        29.0        31.0
## 5        21.5        22.5        23.0        26.5
## 9        23.0        22.5        24.0        27.5
## 13       25.5        27.5        26.5        27.0
## 17       20.0        23.5        22.5        26.0
## 21       24.5        25.5        27.0        28.5
```

Question 1: Approximate Plummer's DIC value for 4 different models of the data.

1a. Approximate Plummer's DIC value for the bivariate formulation

The result can be found at the end of the section. DIC = 405.1

```
library(rjags)
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
### Set up data, initializations, and model

ages <- c(8, 10, 12, 14)

d1 <- list(Y = Y,
          x = ages,
          xbar = mean(ages),
          Omega0 = rbind(c(5, 0),
                         c(0, 0.05)),
          mu0 = c(0,0),
          Sigma0.inv = rbind(c(1.0E-6, 0),
                             c(0, 1.0E-6)))

inits1 <- list(list(tausq.y=1, beta=c(0,0),
```

```r
                    Omega.inv=diag(2)),
               list(tausq.y=100, beta=c(100,100),
                    Omega.inv=100*diag(2)),
               list(tausq.y=0.01, beta=c(-100,-100),
                    Omega.inv=0.01*diag(2)))

m1 <- jags.model("bivariate.model.bug", d1, inits1, n.chains=3)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 108
##    Unobserved stochastic nodes: 30
##    Total graph size: 445
##
## Initializing model
```

```r
### Do a 4000 burn-in
update(m1,  4000)

### Approximate DIC value
dic.samples(model = m1, n.iter=100000, type="pD")
```

```
## Mean deviance:  367
## penalty 38
## Penalized deviance: 405
```

1b. Approximate Plummer's DIC value for the univariate formulation

The result can be found at the end of the section. DIC = 407.8

```r
library(rjags)

### Set up data, initializations, and model

ages <- c(8, 10, 12, 14)

d2 <- list(Y = Y,
          x = ages,
          xbar = mean(ages))

# Choose initialization values for tausq.y, beta1, beta2, sigma.alpha1, and sigma.alpha2
inits2 <- list(list(tausq.y=1, beta1=0, beta2=0,
                    sigma.alpha1=1, sigma.alpha2=1),
               list(tausq.y=100, beta1=100, beta2=100,
                    sigma.alpha1=0.1, sigma.alpha2=0.1),
               list(tausq.y=0.01, beta1=-100, beta2=-100,
                    sigma.alpha1=10, sigma.alpha2=10))
```

```r
# Read in the model in JAGS
m2 <- jags.model("univariate.model.bug", d2, inits2, n.chains=3)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 108
##    Unobserved stochastic nodes: 59
##    Total graph size: 405
##
## Initializing model
```

```r
# Burn in 4000
update(m2, 4000)

### Approximate DIC value
dic.samples(model = m2, n.iter=100000, type="pD")
```

```
## Mean deviance:  369.2
## penalty 38.62
## Penalized deviance: 407.8
```

1c. Modify the univariate formulation.

My JAGS code is below.

```r
# data {
#   dim.Y <- dim(Y)
# }
# model {
#   for(i in 1:dim.Y[1]) {
#
#     for(j in 1:dim.Y[2]) {
#       Y[i,j] ~ dnorm(mu[i,j], tausq.y)
#       mu[i,j] <- alpha1[i] + beta2 * (x[j] - xbar)
#     }
#
#     alpha1[i] ~ dnorm(beta1, 1 / sigma.alpha1^2)
#   }
#
#   tausq.y ~ dgamma(0.001, 0.001)
#   sigma.y <- 1 / sqrt(tausq.y)
#
#   beta1 ~ dnorm(0.0, 1.0E-6)
#   beta2 ~ dnorm(0.0, 1.0E-6)
#   sigma.alpha1 ~ dexp(0.001)
# }
```

Approximate Plummer's DIC value. The result is at the end of the section. DIC = 412.8

```r
### Set up data, initializations, and model

ages <- c(8, 10, 12, 14)

d3 <- list(Y = Y,
          x = ages,
          xbar = mean(ages))

# Choose initialization values for tausq.y, beta1, beta2, sigma.alpha1, and sigma.alpha2
inits3 <- list(list(tausq.y=1, beta1=0, beta2=0,
                   sigma.alpha1=1),
              list(tausq.y=100, beta1=100, beta2=100,
                   sigma.alpha1=0.1),
              list(tausq.y=0.01, beta1=-100, beta2=-100,
                   sigma.alpha1=10))

# Read in the model in JAGS
m3 <- jags.model("univariate.model.modified.c.bug", d3, inits3, n.chains=3)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 108
##    Unobserved stochastic nodes: 31
##    Total graph size: 271
##
## Initializing model
```

```r
update(m3, 4000)

### Approximate DIC value
dic.samples(model = m3, n.iter=100000, type="pD")
```

```
## Mean deviance:  385.5
## penalty 27.31
## Penalized deviance: 412.8
```

1d. Consider just the simple linear regression without any grouping.

JAGS code for model below (shown as comments)

```r
# data {
#   dim.Y <- dim(Y)
# }
# model {
#   for(i in 1:dim.Y[1]) {
#
#     for(j in 1:dim.Y[2]) {
#       Y[i,j] ~ dnorm(mu[i,j], tausq.y)
```

```
#       mu[i,j] <- beta1 + beta2 * (x[j] - xbar)
#     }
#   }
#
#   tausq.y ~ dgamma(0.001, 0.001)
#   sigma.y <- 1 / sqrt(tausq.y)
#
#   beta1 ~ dnorm(0.0, 1.0E-6)
#   beta2 ~ dnorm(0.0, 1.0E-6)
# }
```

Approximate Plummer's DIC value. The result is at the end of the section. DIC = 511.7

```
### Set up data, initializations, and model

ages <- c(8, 10, 12, 14)

d4 <- list(Y = Y,
          x = ages,
          xbar = mean(ages))

# Choose initialization values for tausq.y, beta1, beta2, sigma.alpha1, and sigma.alpha2
inits4 <- list(list(tausq.y=1, beta1=0, beta2=0),
              list(tausq.y=100, beta1=100, beta2=100),
              list(tausq.y=0.01, beta1=-100, beta2=-100))

# Read in the model in JAGS
m4 <- jags.model("univariate.model.modified.d.bug", d4, inits4, n.chains=3)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 108
##    Unobserved stochastic nodes: 3
##    Total graph size: 136
##
## Initializing model
```

```
update(m4, 4000)

### Approximate DIC value
dic.samples(model = m4, n.iter=100000, type="pD")
```

```
## Mean deviance:  508.6
## penalty 3.063
## Penalized deviance: 511.7
```

1e. Compare the Plummer's DIC value of these four models.

The best model according to Plummer's DIC value is the first model, with the lowest DIC value.

Problem 2.

2a. Find expressions for $E(Y_i|\lambda_i)$ and $Var(Y_i|\lambda_i)$

$$E(Y_i|\lambda_i) = \sum_{ally} yf(y|\lambda_i) = \sum_{ally} y\frac{(N_i\lambda_i)^y exp^{-N_i\lambda_i}}{y!} = \sum_{ally} \frac{(N_i\lambda_i)^y exp^{-N_i\lambda_i}}{(y-1)!} = (N_i\lambda_i)exp^{-N_i\lambda_i} \sum_{ally} \frac{(N_i\lambda_i)^{y-1}}{(y-1)!} = (N_i\lambda_i)exp^{-N_i\lambda_i}$$

$$E(Y_i|\lambda_i) = \sum_{ally} yf(y|\lambda_i)$$

$$= \sum_{ally} y\frac{(N_i\lambda_i)^y exp^{-N_i\lambda_i}}{y!}$$

$$= \sum_{ally} \frac{(N_i\lambda_i)^y exp^{-N_i\lambda_i}}{(y-1)!}$$

$$= (N_i\lambda_i)exp^{-N_i\lambda_i} \sum_{ally} \frac{(N_i\lambda_i)^{y-1}}{(y-1)!}$$

$$= (N_i\lambda_i)exp^{-N_i\lambda_i} exp^{N_i\lambda_i}$$

$$= N_i\lambda_i$$

$$Var(Y_i|\lambda_i) = N_i\lambda_i$$

because of Poisson distribution's property.

2b.Present a JAGS model for analyzing the data.

The JAGS model for analyzing the data can be found below (in comments)

```
# model {
#   for(i in 1:length(y)) {
#     y[i] ~ dpois(N[i] * lambda[i])
#     lambda[i] ~ dgamma(alpha, beta)
#     yrep[i] ~ dpois(N[i] * lambda[i])
#   }
#
#   chisq <- sum((y - N*lambda)^2 / (N*lambda))
#   chisqrep <- sum((yrep - N*lambda)^2 / (N*lambda))
#   pb.ind <- chisqrep >= chisq
#
#   alpha ~ dgamma(0.001,0.001)
#   beta ~ dgamma(0.001,0.001)
# }
```

2c. Approximate p_b using JAGS.

The mean of p_b is approximately 0.1753, sd 0.3802. Because the mean of $p_b > 0.05$, we conclude that we do not have evidence against the model and that the prior is a reasonable choice for this dataset. We also see that the estimated mean of chisq and chisqrep are not terribly far away from each other.

```
### Set up data, initializations, and model
library(rjags)

d <- read.table("/Users/gianghale/Desktop/fall-2021/stat-431/airlinerdata.txt", header=TRUE)


inits <- list(list(alpha=1, beta=1, .RNG.seed = 1, .RNG.name="base::Wichmann-Hill"),
              list(alpha=5, beta=5, .RNG.seed = 15, .RNG.name="base::Wichmann-Hill"),
              list(alpha=20, beta=20, .RNG.seed = 30, .RNG.name="base::Wichmann-Hill"))

m <- jags.model("/Users/gianghale/Desktop/fall-2021/stat-431/airliner.posterior.predictive.bug",
                d, inits, n.chains=3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 20
##    Unobserved stochastic nodes: 42
##    Total graph size: 118
##
## Initializing model
```

```
### Make a preliminary run of 1000 iterations, monitoring alpha and beta.

x <- coda.samples(m, c("alpha", "beta"), n.iter=1000)


### Burn-in (2000, after checking for convergence (not shown here))

update(m, 2000)


### Run 100000 more iterations

x <- coda.samples(m, c("chisq","chisqrep","pb.ind"), n.iter=102000)

### Check stats
summary(x)
```

```
##
## Iterations = 4001:106000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 102000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean      SD  Naive SE Time-series SE
```

```
## chisq    32.2099 13.6975 0.0247617      0.0429551
## chisqrep 19.9931  7.3388 0.0132668      0.0133115
## pb.ind    0.1754  0.3803 0.0006875      0.0009223
##
## 2. Quantiles for each variable:
##
##            2.5%   25%   50%   75% 97.5%
## chisq    14.821 23.70 30.06 38.03 61.38
## chisqrep  9.194 14.93 18.90 23.78 36.96
## pb.ind    0.000  0.00  0.00  0.00  1.00
```

Problem 3 (Graduate Section only)

3a. Derive $D(y|\mu)$

Please see the image below for the full derivation. The deviance is defined as twice the negative log likelihood, in other words:

$$D(y|\mu) = -2ln(f(y|\mu)) = -2ln(\frac{1}{\sqrt{2\pi}}exp^{-\frac{(y-\mu)^2}{2}}) = -2(-ln(\sqrt{2\pi}) - \frac{(y-\mu)^2}{2}) = 2ln(\sqrt{2\pi}) + (y-\mu)^2$$

$$\begin{aligned} D(y|\mu) &= -2ln(f(y|\mu)) \\ &= -2ln(\frac{1}{\sqrt{2\pi}}exp^{-\frac{(y-\mu)^2}{2}}) \\ &= -2(-ln(\sqrt{2\pi}) - \frac{(y-\mu)^2}{2}) \\ &= 2ln(\sqrt{2\pi}) + (y-\mu)^2 \end{aligned}$$

3b. Derive $\bar{D}(y) = E(D(y|\mu)|y)$

Please see the image below for the full derivation.

$$\bar{D}(y) = \int D(y|\mu)p(\mu|y)d\mu = \int (2ln(\sqrt{2\pi})+(y-\mu)^2)(\frac{1}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}})d\mu = \int \frac{2ln(\sqrt{2\pi})exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}d\mu + \int \frac{(y-}{}$$

$$\begin{aligned} \bar{D}(y) &= \int D(y|\mu)p(\mu|y)d\mu \\ &= \int (2ln(\sqrt{2\pi}) + (y-\mu)^2)(\frac{1}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}})d\mu \\ &= \int \frac{2ln(\sqrt{2\pi})exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}d\mu + \int \frac{(y-\mu)^2exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}d\mu = \frac{1}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}(\int 2ln(\sqrt{2\pi})exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}d\mu + \int (y-\mu)^2exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}d\mu) \end{aligned}$$

3c. Derive $\hat{D}(y) = D(y|\hat{\mu})$

Please see the image below for the full derivation.

$$\hat{D}(y) = D(y|\hat{\mu}) = -2ln(f(y|\hat{\mu})) = -2ln(\frac{1}{\sqrt{2\pi}}exp^{-\frac{(y-\hat{\mu})^2}{2}}) = -2(-ln(\sqrt{2\pi}) - \frac{(y-\hat{\mu})^2}{2}) = 2ln(\sqrt{2\pi}) + (y - \frac{y}{1+\tau_0^2})^2$$

$$\begin{aligned}\hat{D}(y) = D(y|\hat{\mu}) &= -2ln(f(y|\hat{\mu})) \\ &= -2ln(\frac{1}{\sqrt{2\pi}}exp^{-\frac{(y-\hat{\mu})^2}{2}}) \\ &= -2(-ln(\sqrt{2\pi}) - \frac{(y-\hat{\mu})^2}{2}) \\ &= 2ln(\sqrt{2\pi}) + (y - \frac{y}{1+\tau_0^2})^2\end{aligned}$$

3d. The effective number of parameters

$$p_D = \bar{D} - \hat{D} = \frac{1}{\sqrt{\frac{2\pi}{1+\tau_0^2}}}(\int 2ln(\sqrt{2\pi})exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}d\mu + \int (y-\mu)^2 exp^{-\frac{(y-\mu)^2(1+\tau_0^2)}{2}}d\mu) - (2ln(\sqrt{2\pi}) + (y - \frac{y}{1+\tau_0^2})^2)$$

As τ_0^2 goes to 0, we have

$$p_D = \frac{1}{\sqrt{2\pi}}(\int 2ln(\sqrt{2\pi})exp^{-\frac{(y-\mu)^2}{2}}d\mu + \int (y-\mu)^2 exp^{-\frac{(y-\mu)^2}{2}}d\mu) - 2ln(\sqrt{2\pi})$$