

MTH422A

Vijay Soren (211163)

2024-04-15

1

part(a)

```
#### assgn-5 que1
library(geoR)
library(rjags)
data("gambia")

gam <- gambia

Y <- gambia$pos
X <- as.matrix(gambia[, -3])
X <- scale(X)

data <- list(Y=Y, X=X, n=length(Y), p=dim(X)[2])
model_string <- textConnection("model{
# Likelihood
for(i in 1:n){
  Y[i] ~ dbern(q[i])
  logit(q[i]) <- inprod(X[i,], beta[])
}

#prior
for(j in 1:p){
  beta[j] ~ dnorm(0, 0.01)
}
}")
model <- jags.model(model_string, data=data, n.chains=2, quiet=TRUE)
update(model, 1000, progress.bar="none")
param <- c("beta")
samples <- coda.samples(model, variable.names=param,
  n.iter=5000, progress.bar="none")

summary(samples)
plot(samples)
```

part(b)

```
###que 1 part b
a <- 0.1
b <- 0.1
L <- 65

ar <- as.data.frame(gambia$x)
cont <- table(ar)
s <- rep(1:65, times = c(cont))

data <- list(Y=Y,X=X,s=s, n=length(Y),p=dim(X)[2], L=L, a=a, b=b)
model_string <- textConnection("model{
# Likelihood
for(i in 1:n){
  Y[i] ~ dbern(q[i])

  logit(q[i]) <- inprod(X[i,],beta[]) + alpha[s[i]]
}

#prior
for(j in 1:p){
  beta[j] ~ dnorm(0,0.01)
}
for(j in 1 : L){
  alpha[j] ~ dnorm(0,tau)
}

sigma <- 1/sqrt(tau)
tau ~ dgamma(a,b)

}")

model <- jags.model(model_string, data = data, n.chains = 2,quiet = TRUE)
update(model,1000,progress.bar = "none")
param <- c("beta")
samples <- coda.samples(model, variable.names = param,
                        n.iter = 5000,progress.bar = "none")

summary(samples)

plot(samples)

plot(gambia$x, gambia$y)
```

```
library(MASS)
library(rjags)

# Define the JAGS model
model_string <-textConnection( "
model {
# Likelihood
```

```

for(i in 1:N) {
  Y[i] ~ dnorm(mu[z[i]], tau[z[i]])
  z[i] ~ dcat(p)
}

#priors
for (k in 1:K) {
  mu[k] ~ dnorm(0, 0.0001)
  tau[k] ~ dgamma(0.01, 0.01)

}

# Mixing proportions with uniform Dirichlet prior

p[1:K] ~ ddirch(alpha[1:K])

})")

# Prepare the data
Y <- galaxies
N <- length(Y)

K <- 3
alpha <- rep(1, K)

# Run the model
data <- list(Y = Y, N = N, K = K, alpha = alpha)
model <- jags.model(model_string, data = data,
                    n.chains = 2)

update(model, n.iter = 1000)
samples <- coda.samples(model, variable.names = c("mu","tau","p"),
                        n.iter = 5000, thin = 5)
summary(samples)

# Extract posterior samples
posterior_samples <- as.matrix(samples)
mu <- as.matrix(posterior_samples[,1:3])
tau <- as.matrix(posterior_samples[,7:9])

p <- as.matrix(posterior_samples[,4:6])

S <- 5000

```

```

val <- runif(n = S*351, min = 5000,max = 40000)
y <- matrix(val,nrow = S,ncol = 351)

den <- matrix(0,nrow = S,ncol = 351)

for (i in 1:S) {
  for (j in 1:351) {
    den[i,j] <- p[i,1] *dnorm(y[i,j],mean = mu[i,1],sd = 1/tau[i,1]^2) +
      p[i,2] *dnorm(y[i,j],mean = mu[i,2],sd = 1/tau[i,2]^2) +
      p[i,3] *dnorm(y[i,j],mean = mu[i,3],sd = 1/tau[i,3]^2)
  }
}

```