Advanced Applied Statistics Homework 8 Solutions

Joshua D. Ingram

2022-12-05

As mentioned in class, we are going to try our hands on fitting a logistic regression model not via maximum likelihood (although now you have a deep understanding how that is done), but with the Bayesian approach.

Let's use the Washington Post dataset from the previous HW for it.

Problem 1

Fit the Bayesian Logistic regression model using age as the only predictor. Show the MCMC chains graphically, and the posterior distributions of the intercept and slope. Compare the ML estimators to the Bayesian posterior mean.

Answer: The ML estimators $\hat{\alpha}_{ML} = 0.0040687$ and $\hat{\beta}_{ML} = 0.0057287$ are very similar to the Bayesian posterior means $\hat{\alpha}_{Bayes} = 0.004436559$ and $\hat{\beta}_{Bayes} = 0.005715810$.

```
# logistic regression model
y <- crime$disposition_binary
x <- crime$age
n <- nrow(crime)

# code for the Bayesian model
code <- nimbleCode({

  alpha ~ dnorm(0, sd = 1000)
  beta ~ dnorm(0, sd = 1000)

  for (i in 1:n){

    eta[i] <- alpha + beta * x[i]
    pi[i] <- exp(eta[i]) / (1 + exp(eta[i]))
    y[i] ~ dbern(pi[i])

  }

})

constants <- list(n = n, x = x)
data <- list(y = y)
initial <- list(alpha = mean(y), beta = 0)
Rmodel <- nimbleModel(code, constants, data, initial)
```

```
## Defining model

## Building model

## Setting data and initial values
## Running calculate on model
##   [Note] Any error reports that follow may simply reflect missing values in model variables.

## Checking model sizes and dimensions
```

```r
# visualizing MCMC chain
# Rmodel$plotGraph()
```

```r
conf <- configureMCMC(Rmodel)
```

```
## ===== Monitors =====
## thin = 1: alpha, beta
## ===== Samplers =====
## RW sampler (2)
##   - alpha
##   - beta
```

```r
Rmcmc <- buildMCMC(conf)
Cmodel <- compileNimble(Rmodel)
```

```
## Compiling
##   [Note] This may take a minute.
##   [Note] Use 'showCompilerOutput = TRUE' to see C++ compilation details.
```

```r
Cmcmc <- compileNimble(Rmcmc, project = Rmodel)
```

```
## Compiling
##   [Note] This may take a minute.
##   [Note] Use 'showCompilerOutput = TRUE' to see C++ compilation details.
```

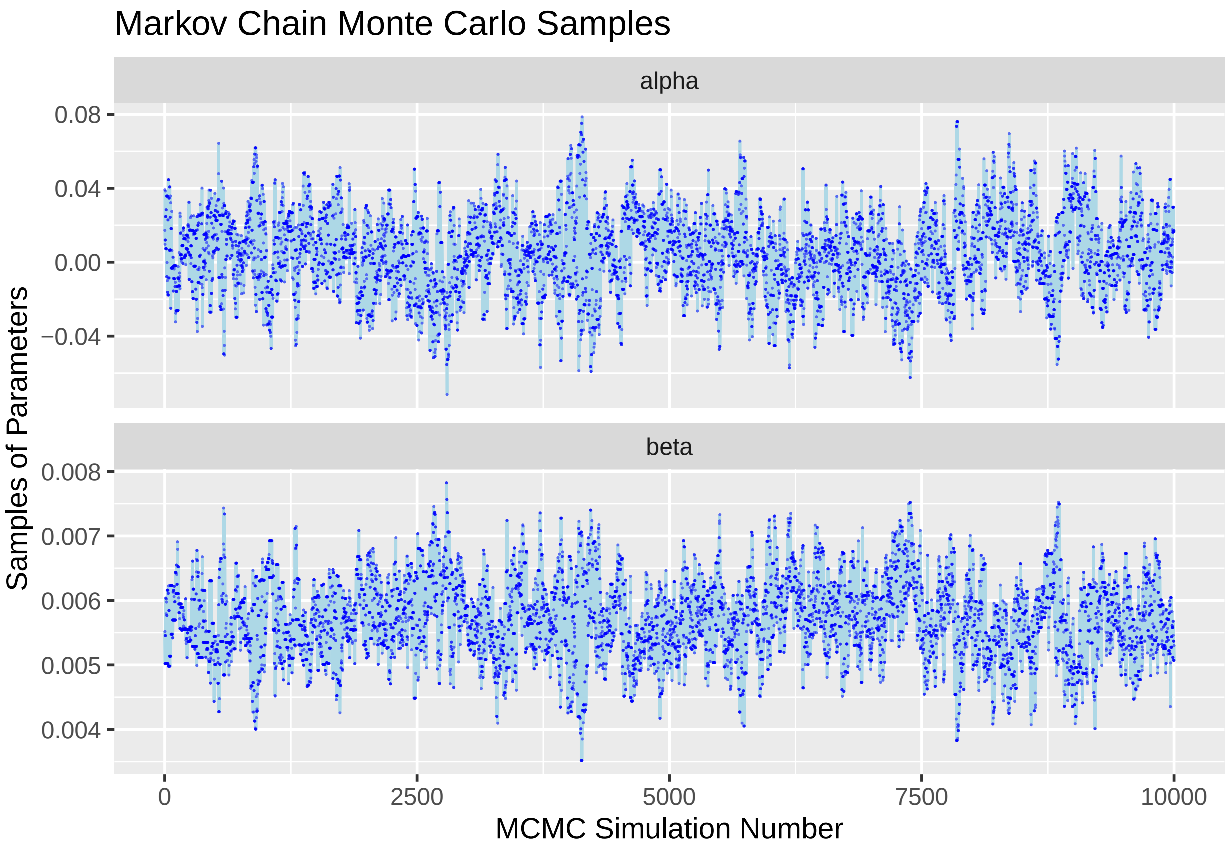
```r
set.seed(22)
results <- runMCMC(Cmcmc, niter = 11000, nburnin = 1000, samples = TRUE, summary = TRUE)
```

```
## running chain 1...

## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

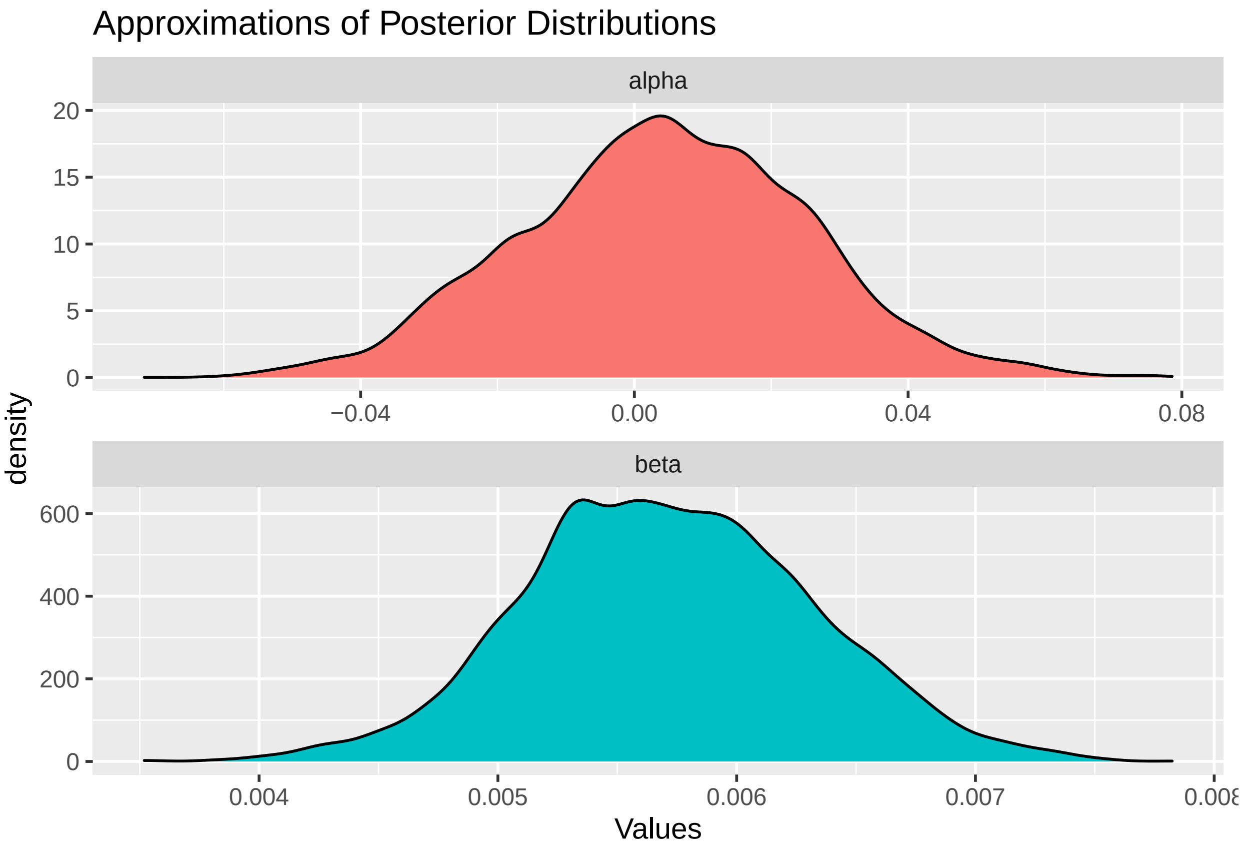
```r
results$summary
```

```
##              Mean      Median      St.Dev.    95%CI_low   95%CI_upp
## alpha 0.004436559 0.004548352 0.0210382079 -0.036306397 0.045316701
## beta  0.005715810 0.005697653 0.0006005335  0.004543837 0.006909582
```

```r
# Visualize MC
df <- as_tibble(results$samples) %>%
  pivot_longer(cols = c("alpha", "beta"), names_to = "Parameter", values_to = "Values") %>%
  add_column(Sequence = rep(1:nrow(results$samples), ncol(results$samples)))

ggplot(data = df, aes(x = Sequence, y = Values)) +
  geom_line(col = "lightblue") +
  geom_point(col = "blue", pch = 16, size = 0.2, alpha = 0.5) +
  facet_wrap(vars(Parameter), nrow = 3, scale = "free_y") +
  labs(x = "MCMC Simulation Number",
       y = "Samples of Parameters",
       title = "Markov Chain Monte Carlo Samples")
```



```
# Visualize Posterior Distribution
ggplot(data = df, aes(x = Values, fill = Parameter)) +
  geom_density(color = "black") +
  facet_wrap(vars(Parameter), nrow = 3, scale = "free") +
  labs(title = "Approximations of Posterior Distributions") +
  theme(legend.position = "none")
```



```r
# Logistic regression model with MLE
fit <- glm(disposition_binary ~ age, family = binomial(link = logit), data = crime)
summary(fit)
```

```
##
## Call:
## glm(formula = disposition_binary ~ age, family = binomial(link = logit),
##     data = crime)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.428  -1.246   1.043   1.111   1.176
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.0040687  0.0219877   0.185    0.853
## age         0.0057287  0.0006325   9.057   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 67755  on 49179  degrees of freedom
## Residual deviance: 67672  on 49178  degrees of freedom
## AIC: 67676
##
## Number of Fisher Scoring iterations: 3
```

Problem 2

Using the Bayesian model above, provide the estimated probability of success (a closed case) at an age of 40 years? Show the entire posterior of the probability of success.

Answer: Predicted probability of success (closed case) at age 40 = 0.555. This is very close to the prediction for the ML logistic regression model at 0.5580417.

```
# logistic regression model
y <- crime$disposition_binary
x <- crime$age
n <- nrow(crime)

# code for the Bayesian model
code <- nimbleCode({

  alpha ~ dnorm(0, sd = 1000)
  beta ~ dnorm(0, sd = 1000)

  for (i in 1:n){

    eta[i] <- alpha + beta * x[i]
    pi[i] <- exp(eta[i]) / (1 + exp(eta[i]))
    y[i] ~ dbern(pi[i])

  }
  yhat ~ dbern( exp(alpha + beta * 40) / (1 + exp(alpha + beta * 40)))
})

constants <- list(n = n, x = x)
data <- list(y = y)
initial <- list(alpha = mean(y), beta = 0)
Rmodel <- nimbleModel(code, constants, data, initial)
```

```
## Defining model

## Building model

## Setting data and initial values

## Running calculate on model
##   [Note] Any error reports that follow may simply reflect missing values in model variables.

## Checking model sizes and dimensions

##   [Note] This model is not fully initialized. This is not an error.
##          To see which variables are not initialized, use model$initializeInfo().
##          For more information on model initialization, see help(modelInitialization).
```

```
conf <- configureMCMC(Rmodel)
```

```
## ===== Monitors =====
## thin = 1: alpha, beta
## ===== Samplers =====
## RW sampler (2)
##   - alpha
##   - beta
## posterior_predictive sampler (1)
##   - yhat
```

```r
conf$addMonitors("yhat")
```

```
## thin = 1: alpha, beta, yhat
```

```r
Rmcmc <- buildMCMC(conf)
Cmodel <- compileNimble(Rmodel)
```

```
## Compiling
##   [Note] This may take a minute.
##   [Note] Use 'showCompilerOutput = TRUE' to see C++ compilation details.
```

```r
Cmcmc <- compileNimble(Rmcmc, project = Rmodel)
```

```
## Compiling
##   [Note] This may take a minute.
##   [Note] Use 'showCompilerOutput = TRUE' to see C++ compilation details.
```

```r
set.seed(22)
results <- runMCMC(Cmcmc, niter = 11000, nburnin = 1000, samples = TRUE, summary = TRUE)
```

```
## running chain 1...
```

```
## |-------------|-------------|-------------|-------------|
## |-------------------------------------------------------|
```

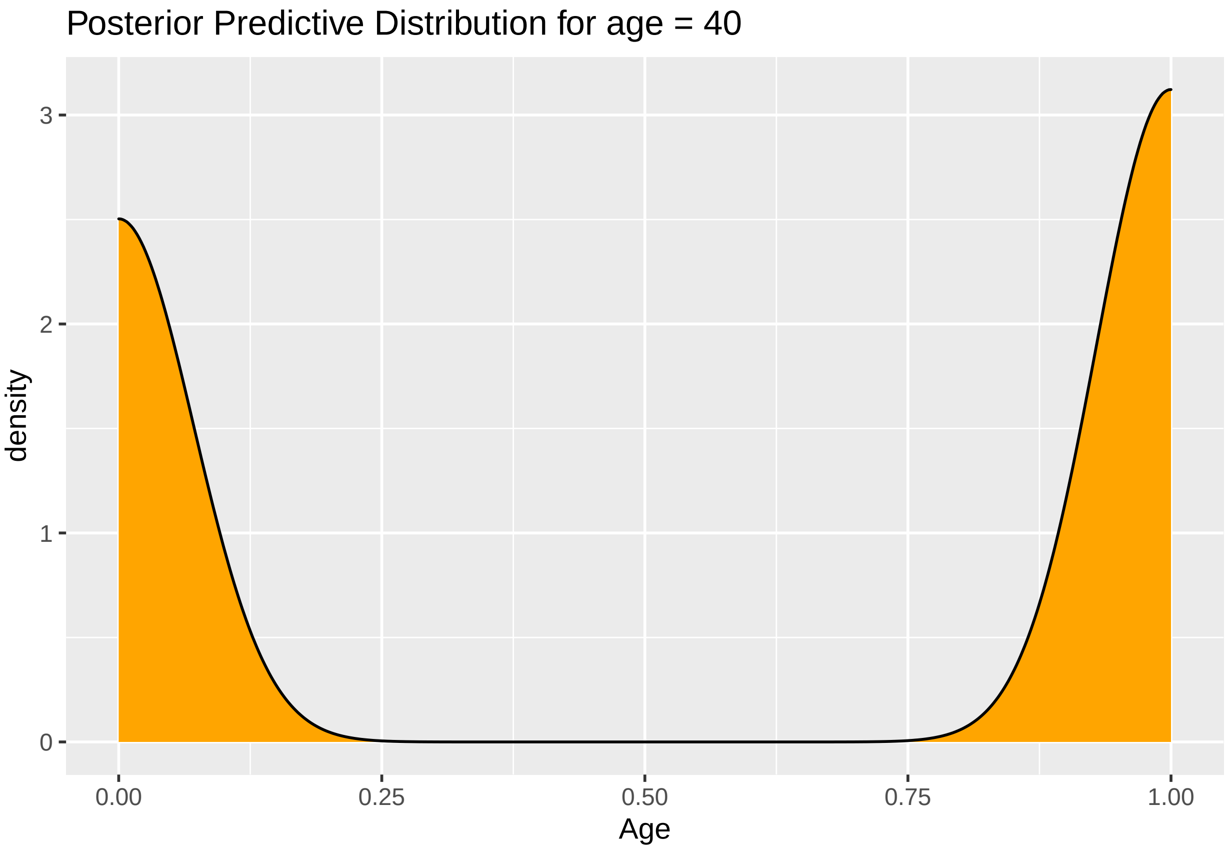
```r
results$summary
```

```
##              Mean      Median      St.Dev.    95%CI_low   95%CI_upp
## alpha 0.003088657 0.003080454 0.0226415668 -0.041534487 0.046046529
## beta  0.005761280 0.005751501 0.0006559879  0.004504027 0.007058579
## yhat  0.555000000 1.000000000 0.4969906437  0.000000000 1.000000000
```

```r
predict(fit, newdata=data.frame(age = 40), type = "response", interval="prediction")
```

```
##         1
## 0.5580417
```

```r
df <- as.data.frame(results$samples)

## Posterior distribution for prediction at age 40
ggplot(data = df, aes(x=yhat)) +
  geom_density(color="black", fill="orange") +
  labs(title = "Posterior Predictive Distribution for age = 40",
       x = "Age")
```


Posterior Predictive Distribution for age = 40
density
3
2
1
0
0.00
0.25
0.50
0.75
1.00
Age