Chapter 12.3c Inference

Jim Albert and Monika Hu

Chapter 12 Bayesian Multiple Regression and Logistic Models

Inference using MCMC

- Fitting a logistic model with a single explanatory variable and a conditional means prior on β.
- Once the prior on the regression coefficients is defined, it is straightforward to simulate from the Bayesian logistic model by MCMC and the JAGS software.

The JAGS script

- The first step is writing a JAGS script defining the logistic regression model including the prior.

```
modelString <-"
model {
## sampling
for (i in 1:N){
   y[i] ~ dbern(p[i])
   logit(p[i]) <- beta0 + beta1*x[i]
}
## priors
beta1 <- (logit(p1) - logit(p2)) / (x1 - x2)
beta0 <- logit(p1) - beta1 * x1
p1 ~ dbeta(a1, b1)
p2 ~ dbeta(a2, b2)
}
```

Comments on JAGS script

- In the sampling section of the script, the loop goes from `1` to `N`, where `N` is the number of observations.
- One uses `dbern()` to denote the Bernoulli response.
- The `logit()` function is written for establishing this linear relationship.
- In the prior section, one expresses `beta0` and `beta1` in terms of `p1`, `p2`, `x1`, and `x2`. One also assigns Beta priors to `p1` and `p2` using the `dbeta()` function.

Define the data and prior parameters

- In the R script , a list `the_data` contains the vector of binary labor participation status values, the vector of family incomes (in $1000), and the number of observations.
- It also contains the shape parameters for the Beta priors on p_1^* and p_2^* and the values of the two incomes, x_1^* and x_2^*.

```
y <- as.vector(LaborParticipation$Participation)
x <- as.vector(LaborParticipation$FamilyIncome)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "a1" = 2.52, "b1" = 20.08,
                 "a2" = 20.59, "b2" = 9.01,
                 "x1" = 20, "x2" = 80)
```

Generate samples from the posterior distribution

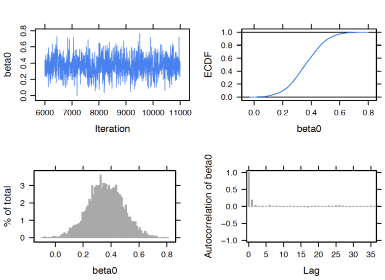
- The `run.jags()` function generates posterior samples by the MCMC algorithm.
- The script below runs one MCMC chain with an adaption period of 1000 iterations, a burn-in period of 5000 iterations and an additional set of 5000 iterations to be collected.
- By specifying `monitor = c("beta0", "beta1")`, one collects values of the regression coefficients.

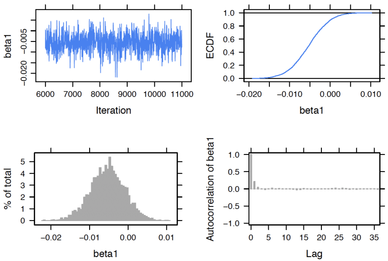
```
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1"),
      adapt = 1000, burnin = 5000, sample = 5000)
```

MCMC diagnostics and summarization

- One applies several diagnostic procedures to check if the simulations appear to converge to the posterior distribution.
- Figures on the next slide display MCMC diagnostic plots for the regression parameters β_0 and β_1.
- From viewing these graphs, it appears that there is a small amount of autocorrelation in the simulated draws and the draws appear to have converged to the posterior distributions.

MCMC diagnostics plots





Posterior Summaries

- By use of the `print()` function, posterior summaries are displayed for the regression parameters.
- From the output, one sees that the posterior 90% interval estimate for the regression slope is $(-0.0143, 0.0029)$.
- There is a negative relationship between family income and labor participation – wives from families with larger income (exclusive of the wife's income) tend not to work. But this relationship does not appear to be strong since the value 0 is included in the 90% interval estimate.

```
print(posterior, digits = 3)
      Lower95   Median Upper95     Mean      SD Mode
beta0   0.101    0.358    0.59     0.36   0.125   --  0
beta1 -0.0143 -0.00524 0.00285 -0.00532 0.00438   -- 7.
```

Learning about probabilities

- One difficulty in interpreting a logistic regression model is that the linear component $\beta_0 + \beta_1 x$ is on the logit scale.
- It is easier to understand when one expresses the fitted model in terms of the probability of participation p_i:

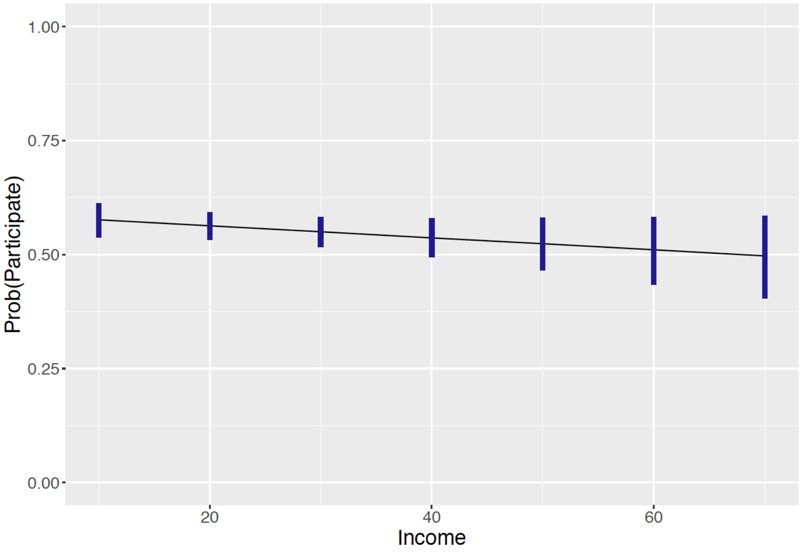
$$p_i = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}.$$

- It is straightforward to simulate the posterior distribution of the probability p_i for fixed x_i. If $(\beta_0^{(s)}, \beta_1^{(s)})$ represents a simulated draw from the posterior of β then $p_i^{(s)}$ is a simulated draw from the posterior of p_i.

Back to Example

- This process was used to obtain simulated samples from the posterior distribution of the probability p_i for the income variable values 10, 20, . . . , 70.
- Figure on the next slide displays posterior medians and 90% interval estimates of the probabilities p_i are displayed
- The takeaway message from this figure is that the probability of labor participation is close to one-half and this probability slightly decreases as the family income increases.

Posterior interval estimates for probability of participation



Prediction

- A related problem is to predict the fraction of labor participation for a sample of n women with a specific family income.
- If $\tilde{y}_i$ represents the number of women who work among a sample of n with family income x_i, one is interested in the posterior predictive distribution of $\tilde{y}_i/n$.
- One represents this predictive density of $\tilde{y}_i$ as

$$f(\tilde{Y}_i = \tilde{y}_i \mid y) = \int \pi(\beta \mid y) f(\tilde{y}_i, \beta) d\beta,$$

where $\pi(\beta \mid y)$ is the posterior density of $\beta = (\beta_0, \beta_1)$ and $f(\tilde{y}_i, \beta)$ is the Binomial sampling density.

Simulating the predictive density

- Suppose that one focuses value x_i^* and one wishes to consider a future sample of $n = 50$. The simulated draws from the posterior distribution of β are stored in a matrix `post`.
- For each of the simulated parameter draws, one computes the probability of labor participation $p^{(s)}$.
- Given those probability values, one simulates Binomial samples where the probability of successes are given by the simulated $\{p^{(s)}\}$, and by dividing $\tilde{y}$ by n, one obtains simulated proportions.
- Each group of simulated draws from the predictive distribution of the labor proportion is summarized by the median, 5th, and 95th percentiles.

R Script

- The function `prediction_interval()`obtains the quantiles of the prediction distribution of $\tilde{y}/n$ for a fixed income level
- The `sapply()` function computes these predictive quantities for a range of income levels.

```
prediction_interval <- function(x, post, n = 20){
    lp <- post[, 1] + x * post[, 2]
    p <- exp(lp) / (1 + exp(lp))
    y <- rbinom(length(p), size = n, prob = p)
    quantile(y / n,
             c(.05, .50, .95))
  }
out <- sapply(seq(10, 70, by = 10),
                prediction_interval, post, n = 50)
```

Figure of Prediction Intervals

- Figure graphs the predictive median and interval bounds against the income variable.
- Note that one is much more certain about the probability of labor participation than the fraction of labor participation in a future sample of 50.