Chapter 12.2b Weakly informative priors and inference through MCMC

Jim Albert and Monika Hu

Chapter 12 Bayesian Multiple Regression and Logistic Models

Priors

- In situations where the data analyst has limited prior information, one can assign a weakly informative prior that has little impact on the posterior.
- Assuming independence, the prior density for the set of parameters $(\beta_0, \beta_1, \beta_2, \sigma)$ is written as a product of the component densities:
- Each of the regression parameters β_0, β_1, and β_2 is assigned a normal density with mean 0 and standard deviations large, say 20.
- One assigns the precision $\phi = 1/\sigma^2$ a Gamma prior with small values of the shape α and the rate parameter β.

MCMC fitting using JAGS

- One uses JAGS to draw MCMC samples from this multiple linear regression model. The process of using JAGS mimics the general approach used in earlier chapters.
- First step in using JAGS describes the multiple regression model by a script.

JAGS Script

```
modelString <-"
model {
## sampling
for (i in 1:N){
   y[i] ~ dnorm(beta0 + beta1*x_income[i] +
                beta2*x_rural[i], invsigma2)
}
## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
beta2 ~ dnorm(mu2, g2)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

Comments

- In the sampling section , the iterative loop goes from `1` to `N`, where `N` is the number of observations with index `i`.
- Recall that `dnorm` in JAGS represents a normal distribution in terms of the mean and the precision parameter `invsigma2`.
- The variable `sigma` is defined so one can track the simulated values of the standard deviation σ.
- The variables `m0`, `m1`, `m2` correspond to the means, and `g0`, `g1`, `g2` correspond to the precisions of the Normal prior densities for the three regression parameters.

Define the data and prior parameters

- In the R script below, a list `the_data` contains the vector of log expenditures, the vector of log incomes, the indicator variables for the categories of the binary categorical variable, and the number of observations.
- This list also contains the means and precisions of the Normal priors for `beta0` through `beta2` and the values of the two parameters `a` and `b` of the Gamma prior for `invsigma2`.
- Since our prior is weakly informative, the regression coefficients have small precision values and the Gamma prior parameter values are small.

Define the data and prior parameters

```
y <- as.vector(CEsample$log_TotalExp)
x_income <- as.vector(CEsample$log_TotalIncome)
x_rural <- as.vector(CEsample$Rural)
N <- length(y)
the_data <- list("y" = y, "x_income" = x_income,
                 "x_rural" = x_rural, "N" = N,
                 "mu0" = 0, "g0" = 0.0025,
                 "mu1" = 0, "g1" = 0.0025,
                 "mu2" = 0, "g2" = 0.0025,
                 "a" = 0.001, "b" = 0.001)
```

Generate samples from the posterior distribution

- The `run.jags()` function in the `runjags` package generates posterior samples.
- 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 20,000 iterations to be run and collected for inference.
- By using the argument `monitor = c("beta0", "beta1", "beta2", "sigma")`, one keeps tracks of all four model parameters. The output variable `posterior` contains a matrix of simulated draws.

Run JAGS function

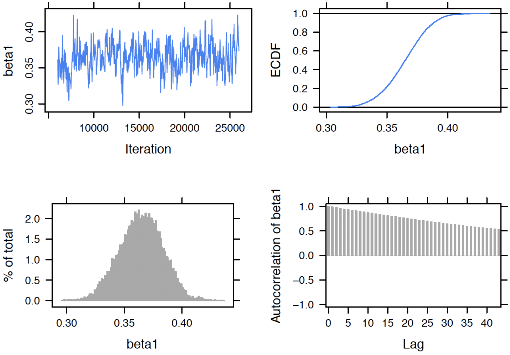
```
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1",
                                  "beta2", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 20000)
```

MCMC diagnostics

- To obtain valid inferences from the posterior draws from the MCMC simulation, one should assess convergence of the MCMC chain.
- The `plot()` function with the argument input `vars` returns four diagnostic plots (trace plot, empirical CDF, histogram and autocorrelation plot) for the specified parameter.

MCMC Diagnostic plots for β_1

```
plot(posterior, vars = "beta1")
```



Comments

- The trace plot shows MCMC mixing for the 20,000 simulated draws of β_1.
- The autocorrelation plot indicates relatively large correlation values between adjacent posterior draws of β_1.
- Since the mixing was not great, it was decided to take a larger sample of 20,000 draws to get good estimates of the posterior distribution.

Summarization of the posterior

- Posterior summaries of the parameters are obtained by use of the `print()` function.

```
print(posterior, digits = 3)
      Lower95 Median Upper95   Mean     SD Mode    MCe
beta0    4.59   4.95    5.36   4.95  0.201   --   0.01
beta1   0.328  0.365     0.4  0.365 0.0188   --  0.001
beta2  -0.482 -0.267 -0.0476 -0.269  0.112   --  0.001
sigma   0.735  0.769   0.802  0.769 0.0172   -- 0.0001
```

Useful Predictors?

- One can determine if the two variables are useful predictors by inspecting the location of the 90% probability intervals
- The interval for β_1 (corresponding to log income) is (0.328, 0.400) and the corresponding interval for β_2 (corresponding to the rural variable) is $(-0.482, -0.048)$.
- Since both intervals do not cover zero, both log income and the rural variables appear helpful in predicting log expenditure.

Posterior of Expected Response

- Suppose one is interested in learning about the expected log expenditure equal to

$$\beta_0 + \beta_1 x_{income}$$

for urban CUs, and equal to

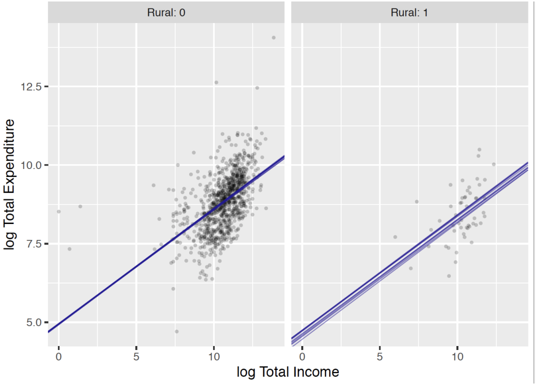
$$\beta_0 + \beta_1 x_{income} + \beta_2$$

for rural CUs.

- Compute these functions on the simulated draws of β to find posteriors of expected response.

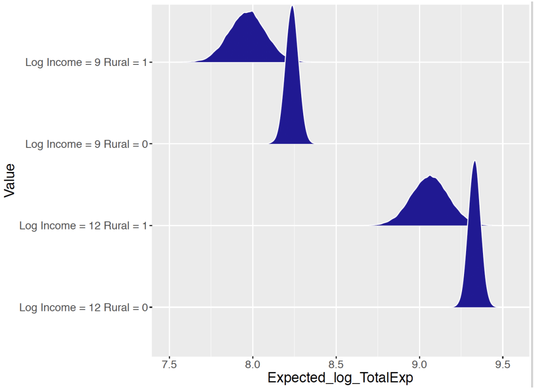
Posterior of Expected Response

- Figure displays simulated draws from the posterior of the expected log expenditure for the urban and rural cases.
- There is more variation in the posterior draws for the rural units.



Posterior Expected Values (continued)

- Figure displays the posterior density of the mean log expenditure for four values of the predictors.
- It is pretty clear from this graph that log income is the more important predictor.



Prediction

- Want to predict a CU's log expenditure for a particular set of predictor values.
- Let $\tilde{Y}$ denote the future response value for the expenditure for given values of income x^*_{income} and rural value x^*_{rural}.
- One represents the posterior predictive density of $\tilde{Y}$ as

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

  where $\pi(\beta, \sigma | y)$ is the posterior density and $f(\tilde{Y} = \tilde{y} \mid y, \beta, \sigma)$ is the Normal sampling density.

R Work

To simulate from the posterior predictive distribution ...

- One simulates a single draw from $f(\tilde{Y} = \tilde{y} \mid y)$ by first simulating a value of (β, σ) from the posterior – call this draw $(\beta^{(s)}, \sigma^{(s)})$.
- One simulates a draw of $\tilde{Y}$ from a Normal density with mean $\beta_0^{(s)} + \beta_1^{(s)} x^*_{income} + \beta_2^{(s)} x^*_{rural}$ and standard deviation $\sigma^{(s)}$.

Repeat this process for a large number of iterations to get a sample from the posterior predictive distribution of $\tilde{Y}$.

R function

- The function `one_predicted()` simulates a sample from the posterior prediction distribution for particular predictor values x^*_{income} and x^*_{rural}.

```
one_predicted <- function(x1, x2){
  lp <- post[ , "beta0"] +  x1 * post[ , "beta1"] +
    x2 * post[, "beta2"]
  y <- rnorm(5000, lp, post[, "sigma"])
  data.frame(Value = paste("Log Income =", x1,
                           "Rural =", x2),
             Predicted_log_TotalExp = y)
}
df <- map2_df(c(12, 12),
              c(0, 1), one_predicted)
```

Example

- This procedure is implemented for two pairs of predictor values (log income, rural).
- Figure displays density estimates of the posterior predictive distributions for the two cases. (Compare with the posterior density estimates for the expected response.)