GPH-GU2372/3372
Applied Bayesian Analysis in Public Health

Review Session

Hai Shu, PhD

12/12/2022

Bayes' theorem

- Bayes' theorem for density functions:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int_{\theta\in\Theta} p(y|\theta)p(\theta)d\theta} \propto p(y|\theta)p(\theta)$$

Hierarchical models

$$\begin{cases} Y_{1,j}, Y_{2,j}, \ldots, Y_{n_j,j} | \phi_j \overset{iid}{\sim} p(y|\phi_j) \qquad j = 1, \ldots, m & (1) \\ \phi_1, \ldots, \phi_m | \psi \overset{iid}{\sim} p(\phi|\psi) & (2) \\ \psi \sim p(\psi) & (3) \end{cases}$$

- The first stage model, model (1), models the variability of the observations within a group; the second stage model, model (2), models the variability across groups; and the third stage model, model (3), provides the distribution function for the parameter ψ.
- The two distribution in (1) and (2) are sampling distributions while the distribution in (3) is a prior distribution.
- The prior distribution $p(\psi)$ might depend on other parameters, say γ_ψ. Those are called hyperparameters. We can provide a prior distribution for γ_ψ: this would be called a hyperprior.

The hierarchical normal model with same σ^2

$$
\begin{aligned}
y_{1,j}, \ldots, y_{n_j,j} | \phi_j = (\theta_j, \sigma^2) &\overset{iid}{\sim} N(\theta_j, \sigma^2) \\
\theta_1, \theta_2, \ldots, \theta_m &\perp \sigma^2 \\
\theta_1, \theta_2, \ldots, \theta_m | \psi = (\mu, \tau^2) &\overset{iid}{\sim} N(\mu, \tau^2) \\
\sigma^2 &\sim \text{InverseGamma}(\tfrac{\nu_0}{2}, \tfrac{\nu_0 \sigma_0^2}{2}) \\
\mu &\sim N(\mu_0, \gamma_0^2) \\
\tau^2 &\sim \text{InverseGamma}(\tfrac{\eta_0}{2}, \tfrac{\eta_0 \tau_0^2}{2})
\end{aligned}
$$

- See the R code in Lecture8_code.R

The hierarchical normal model with unequal σ_j^2's

$$
\begin{aligned}
y_{1,j},\ldots,y_{n_j,j}|\theta_j,\sigma_j^2 &\overset{iid}{\sim} N(\theta_j,\sigma_j^2) && j=1,\ldots,m\\
\theta_j &\perp \sigma_j^2 && j=1,\ldots,m\\
\theta_1,\ldots,\theta_m|\mu,\tau^2 &\overset{iid}{\sim} N(\mu,\tau^2)\\
\sigma_1^2,\ldots,\sigma_m^2 &\overset{iid}{\sim} \text{InverseGamma}(\tfrac{\nu_0}{2},\tfrac{\nu_0\sigma_0^2}{2})\\
\mu &\sim N(\mu_0,\gamma_0^2)\\
\tau^2 &\sim \text{InverseGamma}(\tfrac{\eta_0}{2},\tfrac{\eta_0\tau_0^2}{2})\\
\nu_0 &\sim p(\nu_0)\\
\sigma_0^2 &\sim p(\sigma_0^2)
\end{aligned}
$$

- We can use a Gamma(a,b) prior on σ_0^2.
- Finding a prior on ν_0 is more difficult, as no distribution will be conjugate.
  We choose to use a discrete prior. The book uses a geometric prior $p(\nu_0) \propto \exp(-\alpha\nu_0)$ on the set of integers $\{1,2,\ldots,\}$.
  For simplicity, we may use a discrete uniform distribution $p(\nu_0) = \frac{1}{30}$ on the set $\{1,2,\ldots,30\}$.

Linear regression model

$$Y_i = \beta' \mathbf{x}_i + \varepsilon_i \qquad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

for each $i = 1, \ldots, n$ implies that $Y_i | \beta, \mathbf{x}_i, \sigma^2 \sim N(\beta' \mathbf{x}_i, \sigma^2)$.

- Let's consider the case where
  $p(\beta, \sigma^2) = p(\beta) \cdot p(\sigma^2) = N_p(\beta_0, \mathbf{\Sigma}_0) \cdot \text{InverseGamma}(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2})$.
  - If we have raw data $D_0 = (n_0, \mathbf{y}_0, \mathbf{X}_0)$ from a similar study we can determine $\beta_0, \mathbf{\Sigma}_0, \nu_0, \sigma_0^2$ as follows:
    - we can set β_0 equal to the OLS estimate $\hat{\beta}_{0,OLS}$ of β in a linear regression for $\mathbf{y}_0$ on $\mathbf{X}_0$.
    - we can set $\mathbf{\Sigma}_0$ equal to $a_0^{-1}(\mathbf{X}_0' \mathbf{X}_0)^{-1}$ for some predetermined constant a_0
    - we can set σ_0^2 equal to:

$$\sigma_0^2 = \frac{1}{c_0(n_0 - p)} \left(\mathbf{y}_0 - \mathbf{X}_0 \hat{\beta}_{0,OLS}\right)' \left(\mathbf{y}_0 - \mathbf{X}_0 \hat{\beta}_{0,OLS}\right)$$

      for some constant c_0
    - we can set ν_0 equal to some predetermined constant c_0.

(Compare with unit information prior (textbook p.156) and g-prior (p.157))

Linear regression model (`MCMCpack`)

https://cran.r-project.org/web/packages/MCMCpack/MCMCpack.pdf

Usage

```
MCMCregress(
  formula,
  data = NULL,
  burnin = 1000,
  mcmc = 10000,
  thin = 1,
  verbose = 0,
  seed = NA,
  beta.start = NA,
  b0 = 0,
  B0 = 0,
  c0 = 0.001,
  d0 = 0.001,
  sigma.mu = NA,
  sigma.var = NA,
  marginal.likelihood = c("none", "Laplace", "Chib95"),
  ...
)
```

The model takes the following form:

$$y_i = x_i'\beta + \varepsilon_i$$

Where the errors are assumed to be Gaussian:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

We assume standard, semi-conjugate priors:

$$\beta \sim \mathcal{N}(b_0, B_0^{-1})$$

And:

$$\sigma^{-2} \sim \mathcal{G}amma(c_0/2, d_0/2)$$

Generalized linear regression models (Metropolis algorithm)

- Poisson linear regression (`MCMCpoisson()`):

$$Y_i|\theta_i \sim \text{Poisson}(\theta_i)$$

$$\log(\theta_i) = \boldsymbol{\beta}'\boldsymbol{x}_i \quad \text{i.e.} \quad \theta_i = \exp(\boldsymbol{\beta}'\boldsymbol{x}_i)$$

$$\boldsymbol{\beta} \sim N_p(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$$

- Logistic regression (`MCMClogit()`):

$$Y_i|\pi_i \sim \text{Bernoulli}(\pi_i)$$

$$\text{logit}(\pi_i) = \log(\frac{\pi_i}{1-\pi_i}) = \boldsymbol{\beta}'\boldsymbol{x}_i \quad \text{i.e.} \quad \pi_i = \frac{\exp(\boldsymbol{\beta}'\boldsymbol{x}_i)}{1+\exp(\boldsymbol{\beta}'\boldsymbol{x}_i)}$$

$$\boldsymbol{\beta} \sim N_p(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$$

Hierarchical regression models

- Collecting all the observations for students in the same school in a $n_j \times 1$ vector $\mathbf{Y}_j$ and collecting all the covariate information for students in school j into a $n_j \times p$ matrix $\mathbf{X}_j$, we have that the first stage of the hierarchical linear regression model is

$$\mathbf{Y}_j = \mathbf{X}_j \beta_j + \varepsilon_j$$

$$\implies \quad \mathbf{Y}_j | \mathbf{X}_j, \beta_j, \sigma^2 \sim N_{n_j}(\mathbf{X}_j \beta_j, \sigma^2 \mathbf{I}_{n_j})$$

  for $j = 1, \ldots, m$ where $\varepsilon_j = (\varepsilon_{1,j} \ldots \varepsilon_{n_j,j})'$.

- At the second stage of the model, we want to allow for sharing of information across groups in the regression coefficients. Thus we model:

$$\beta_1, \beta_2, \ldots, \beta_m | \theta, \mathbf{\Sigma} \overset{iid}{\sim} N_p(\theta, \mathbf{\Sigma})$$

- This model captures the variability between groups in the β_j's.

Hierarchical regression models

- Note that if $\mathbf{Z} \sim N_p(\mu, \mathbf{\Lambda})$, we can express $\mathbf{Z}$ as

$$\mathbf{Z} = \mu + \xi \qquad \xi \sim N_p(\mathbf{0}, \mathbf{\Lambda})$$

- Applying this to the β_j for $j = 1, \ldots, m$, we have

$$\beta_j = \theta + \xi_j \qquad \xi_j \overset{iid}{\sim} N_p(\mathbf{0}, \mathbf{\Sigma})$$

- The operation or rewriting $\beta_j | \theta, \mathbf{\Sigma} \sim N_p(\theta, \mathbf{\Sigma})$ as

$$\begin{array}{rcl} \beta_j & = & \theta + \xi_j \\ \xi_j | \mathbf{\Sigma} & \overset{iid}{\sim} & N_p(\mathbf{0}, \mathbf{\Sigma}) \end{array}$$

is called hierarchical centering and is one of the suggested methods to improve convergence in MCMC algorithms.

Hierarchical regression models

- Substituting the expression of β_j as $\theta + \xi_j$ into the linear regression model for $\mathbf{Y}_j$, we obtain

$$\begin{aligned} \mathbf{Y}_j &= \mathbf{X}_j\beta_j + \varepsilon_j \\ &= \mathbf{X}_j(\theta + \xi_j) + \varepsilon_j \\ &= \mathbf{X}_j\theta + \mathbf{X}_j\xi_j + \varepsilon_j \end{aligned} \tag{1}$$

- Since in this parametrization, θ is not a random variable, θ represent the fixed effect of the p covariates on the response. θ refers to the entire population.
- On the other hand, as for each $j = 1, \ldots, m$, ξ_j is a random variable, ξ_j represent the school-specific random effects of the p covariates on the response for school j.
  The ξ_j, $j = 1, \ldots, m$, represent the deviation from the overall population effect θ for each group j.
- In light of this, the model (1) above is called a linear mixed model.

Hierarchical regression models

Model:

$$\begin{aligned}\mathbf{Y}_j &= \mathbf{X}_j\boldsymbol{\beta}_j + \boldsymbol{\varepsilon}_j \\ &= \mathbf{X}_j(\boldsymbol{\theta} + \boldsymbol{\xi}_j) + \boldsymbol{\varepsilon}_j \\ &= \mathbf{X}_j\boldsymbol{\theta} + \mathbf{X}_j\boldsymbol{\xi}_j + \boldsymbol{\varepsilon}_j, \qquad \boldsymbol{\varepsilon}_j \sim N_{n_j}(\mathbf{0}, \sigma^2\mathbf{I}_{n_j})\end{aligned}$$

$$\boldsymbol{\beta}_j = \boldsymbol{\theta} + \boldsymbol{\xi}_j | \boldsymbol{\theta}, \boldsymbol{\Sigma} \overset{iid}{\sim} N_p(\boldsymbol{\theta}, \boldsymbol{\Sigma}) \quad \Leftrightarrow \quad \boldsymbol{\xi}_j \overset{iid}{\sim} N_p(\mathbf{0}, \boldsymbol{\Sigma})$$

Priors:

$$\sigma^2 \sim \text{InverseGamma}(\tfrac{\nu_0}{2}, \tfrac{\nu_0\sigma_0^2}{2})$$

$$\boldsymbol{\theta} \sim N_p(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0)$$

$$\boldsymbol{\Sigma} \sim \text{InverseWishart}(\boldsymbol{\eta}_0, \mathbf{S}_0^{-1})$$

Hierarchical regression models (MCMCpack)

Usage

```
MCMChregress(
  fixed,
  random,
  group,
  data,
  burnin = 1000,
  mcmc = 10000,
  thin = 10,
  verbose = 1,
  seed = NA,
  beta.start = NA,
  sigma2.start = NA,
  Vb.start = NA,
  mubeta = 0,
  Vbeta = 1e+06,
  r,
  R,
  nu = 0.001,
  delta = 0.001,
  ...
)
```

The model takes the following form:

$$y_i = X_i\beta + W_i b_i + \varepsilon_i$$

Where each group i have k_i observations.

Where the random effects:

$$b_i \sim \mathcal{N}_q(0, V_b)$$

And the errors:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{k_i})$$

We assume standard, conjugate priors:

$$\beta \sim \mathcal{N}_p(\mu_\beta, V_\beta)$$

And:

$$\sigma^2 \sim \mathcal{IG}amma(\nu, 1/\delta)$$

And:

$$V_b \sim \mathcal{IW}ishart(r, rR)$$

Due to different notations of the Inverse Wishart distribution, their rR = our $\mathbf{S}_0$.

Hierarchical generalized linear regression models

- Replace the identity link function by another link function, e.g., the log link function for Hierarchical Poisson linear regression model. Use, e.g., `MCMChpoisson()`; see `lecture10_code_MCMCpack.R`.

(Ordered) probit regression

- If Y can take on K values, say $\{1, 2, \ldots, K\}$, the function g can be described with $K-1$ parameters, $g_1, \ldots, g_{K-1}$ and we set

$$Y = \begin{cases} 1 & \text{if} \quad -\infty = g_0 < Z \leq g_1 \\ 2 & \text{if} \quad g_1 < Z \leq g_2 \\ \vdots & \vdots \quad \vdots \\ K & \text{if} \quad g_{K-1} < Z < g_K = +\infty \end{cases}$$

- We complete the specification of the ordered probit regression model by placing a prior on β and on the thresholds $g_1, \ldots, g_{K-1}$ leading to the following model

$$\begin{aligned} Y_i | Z_i &= g(Z_i) \\ Z_i | \beta &\overset{ind}{\sim} N(\mathbf{X}_i' \beta, 1) \\ \beta &\sim N_p(\beta_0, \mathbf{\Sigma}_0) \\ \mathbf{g} = \{g_1, \ldots, g_{K-1}\} &\sim p(\mathbf{g}) \end{aligned}$$

- Use `MCMCoprobit()` for $K \geq 3$ and `MCMCprobit()` for $K = 2$. See `Lecture11_code_2(MCMCpack).R`

Gibbs sampler

- In general, the Gibbs sampling algorithm works as follows: suppose that we have observations $y_1, \ldots, y_n | \phi \overset{iid}{\sim} p(y|\phi)$ where $\phi = (\phi_1, \ldots, \phi_p)$ and we place a prior $p(\phi)$ on ϕ.
- We are interested in the joint posterior distribution $p(\phi|y_1, \ldots, y_n)$.
- **Given** an initial value, a starting point, $\phi^{(0)} = \left(\phi_1^{(0)}, \phi_2^{(0)}, \ldots, \phi_p^{(0)}\right)$, the Gibbs sampling algorithm provides an approximation to the posterior distribution $p(\phi|y_1, \ldots, y_n)$ by sampling iteratively from the full conditional distributions.
  Precisely, at the s-th iteration, the algorithm proceeds as follows:
  - sample $\phi_1^{(s)}$ from $p(\phi_1|y_1, \ldots, y_n, \phi_2^{(s-1)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)})$
  - sample $\phi_2^{(s)}$ from $p(\phi_2|y_1, \ldots, y_n, \phi_1^{(s)}, \phi_3^{(s-1)}, \ldots, \phi_p^{(s-1)})$
  - sample $\phi_3^{(s)}$ from $p(\phi_3|y_1, \ldots, y_n, \phi_1^{(s)}, \phi_2^{(s)}, \phi_4^{(s-1)}, \ldots, \phi_p^{(s-1)})$
  - $\ldots$
  - sample $\phi_{p-1}^{(s)}$ from $p(\phi_{p-1}|y_1, \ldots, y_n, \phi_1^{(s)}, \phi_2^{(s)}, \phi_3^{(s)}, \ldots, \phi_{p-2}^{(s)}, \phi_p^{(s-1)})$
  - sample $\phi_p^{(s)}$ from $p(\phi_p|y_1, \ldots, y_n, \phi_1^{(s)}, \phi_2^{(s)}, \phi_3^{(s)}, \ldots, \phi_{p-1}^{(s)})$

Metropolis algorithm

- The algorithm works as follows:
  - Choose a number S of iterations.
  - Sample a starting value $\theta^{(0)}$ for which $p(\theta^{(0)}|\mathbf{y}) > 0$ from some starting distribution $p_0(\theta)$. This starting distribution might be an approximation to the posterior distribution $p(\theta|\mathbf{y})$, or $\theta^{(0)}$ might be an initial value close to the mean of $p(\theta|\mathbf{y})$.
  - For iterations $k = 1, \ldots, S$, repeat the following steps
    1. sample a proposal or candidate value $\theta^\star$ from a jumping or proposal distribution $J_k(\theta^\star|\theta^{(k-1)})$.
       For the Metropolis algorithm, the jumping distribution must be symmetric, that is: $J_k(\theta_a|\theta_b) = J_k(\theta_b|\theta_a)$ for **all** θ_b, θ_a and k.
    2. calculate the ratio of densities $r = \frac{p(\theta^\star|\mathbf{y})}{p(\theta^{(k-1)}|\mathbf{y})}$
    3. set $\theta^{(k)}$ equal to $\theta^\star$ with probability equal to $\min(r, 1)$. Otherwise, set $\theta^{(k)}$ equal to $\theta^{(k-1)}$.

Metropolis-Hastings algorithm

- The Metropolis-Hastings algorithm generalizes the basic Metropolis algorithm in two ways:
  1. there is no requirement that the jumping distribution $J_k(\theta^\star|\theta^{(k-1)})$ is symmetric. That is, we no longer require
  $$J_k(\theta_b|\theta_a) = J_k(\theta_a|\theta_b)$$
  2. to correct for asymmetry in the jumping distribution, the ratio r that is used to determine whether to accept or reject the proposed value $\theta^\star$ is replaced by the ratio
  $$r = \frac{\frac{p(\theta^\star|\mathbf{y})}{J_k(\theta^\star|\theta^{(k-1)})}}{\frac{p(\theta^{(k-1)}|\mathbf{y})}{J_k(\theta^{(k-1)}|\theta^\star)}} = \frac{p(\theta^\star|\mathbf{y}) \cdot J_k(\theta^{(k-1)}|\theta^\star)}{p(\theta^{(k-1)}|\mathbf{y}) \cdot J_k(\theta^\star|\theta^{(k-1)})}$$

  Note that the ratio r is always defined because a jump from $\theta^{(k-1)}$ to $\theta^\star$ can only occur if both $p(\theta^{(k-1)}|\mathbf{y})$ and $J_k(\theta^\star|\theta^{(k-1)})$ are nonzero.

Gibbs sampler + Metropolis algorithm

- The Gibbs sampler and the Metropolis algorithm can be combined together. If the parameter on which we are making inference is a multidimensional vector $\theta = (\theta_1, \ldots, \theta_p)$ and some of the full conditionals have closed forms while some have not, then we can proceed as follows:
  - sample the parameters for which we can sample directly from the full conditional using a Gibbs sampler
  - update the parameters for the full conditional is not available in closed form using a Metropolis or Metropolis-Hastings algorithm