Intro to Bayesian Computing Series (I)

IBC members present

Yi He, William Wu, Cindy Chen, Chuan Zhou, Heidi, Angel An, & Lily Wang

Yi.He@Vanderbilt.Edu

Department of Biostatistics

School of Medicine

Vanderbilt University

Feburary 16, 2006

A little history of statistics

As Efron (2004) said,

- 19th century is Bayesian statistics
- 20th century is frequentist statistics
- What's next? Possible Empirical Bayesian?
- In the last two decades, fast development of computing faciliaties and invention of Markov Chain Monte Carlo (MCMC) faciliates Bayesian analysis.
- Bayesian analysis now is feasible and attracts scientists more and more attention in various applications.

Baye's rule

- The essence of Bayesian analysis is to draw inference of unknow quantities or quantiles of interest from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$, which is from prior beliefs and data information. Bayes' rule provides such connection.

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}) &= \frac{p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})}{p(\mathbf{y})} \\ \text{posterior} &\propto p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta}) \propto \text{prior} \times \text{data information} \end{aligned} \quad (1)$$

- Why this makes sense? Our human brain is essentially a Bayesian machine.

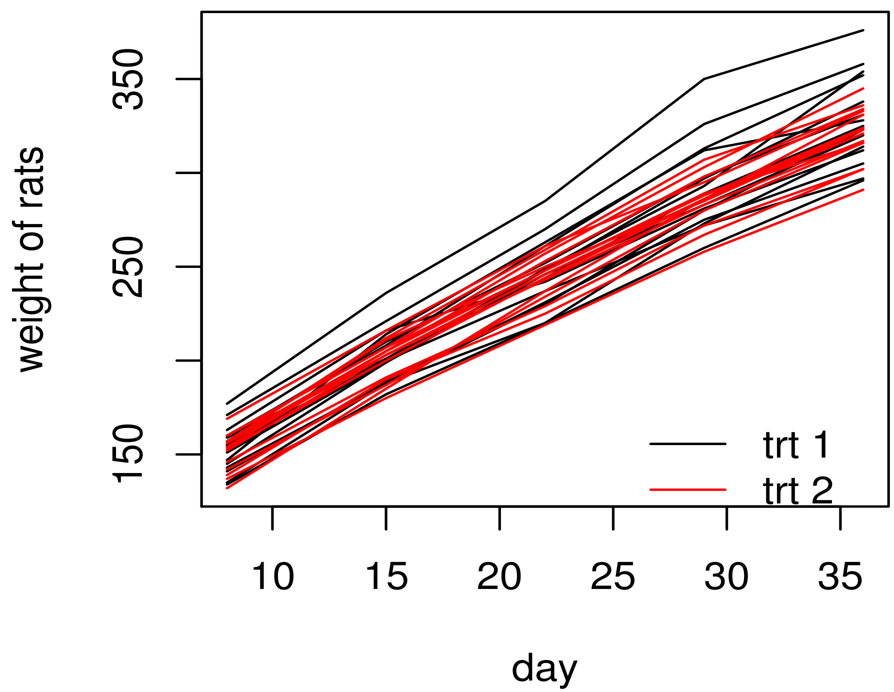
Non-informative prior

Bayesian analysis requires prior information. Can I still use Bayesian analysis without “prior" information about $\boldsymbol{\theta}$? Yes.

- Non-informative prior, vague prior, reference prior
- Ways to construct non-informative prior
  - Intuitively, flat / almost flat over the parameter space. e.g., $X_i \sim N(\mu, \sigma^2), iid$ with σ^2 known. Then use prior $p(\mu) \propto 1$ or $p(\mu) \sim N(0, 10^6)$.
  - Jeffrey's prior, which is invariant under transformation, $p(\boldsymbol{\theta}) \propto [I(\boldsymbol{\theta})]^{1/2}$ where $I(\boldsymbol{\theta})$ is the expected Fisher information in the model.
  - Non-informative prior may be improper, in the sense that $\int p(\theta) d\theta = \infty$.

Rats Data

Data are obtained from WinBUGS (Spielhalter et al. 2002) example volume I (`http://www.mrc-bsu.cam.ac.uk/bugs`), originally from Gelfand et al. (1990).



Random effects model

- The data suggest a growing pattern with age with a little downward curvature.

$$
\begin{aligned}
Y_{ij} &\sim N(a_i + \beta trt_i + b_i(x_j - \bar{x}), \tau_0^{-1}) \\
a_i &\sim N(\mu_a, \tau_a^{-1}) \\
b_i &\sim N(\mu_b, \tau_b^{-1}),
\end{aligned} \tag{2}
$$

  where $\bar{x} = 22$, the average of x, trt_i is the group assignment for rat i, and τ_0, τ_a, τ_b are precisions (1/variance) for the corresponding normal distributions.

- This model suggests that for each subject (i.e., fix random effects a_i and b_i, and group trt_i), the growth curve is linear with noise precision τ_0. The group effect can be captured by β.

Hierarchical normal / normal model

- Hierarchical normal / normal model is analogous to mixed model, however in Bayesian world, there are no fixed effects because all parameters are treated as random with distributions.
- The above model is not a fully Bayesian model, because it can be treated as a typical mixed model with fixed effects `intercept`, `day`, `trt` and random effects `intercept`, `day`.
- The first equation in model (2) specifies the likelihood and the other two specify priors for $\boldsymbol{a}$ and $\boldsymbol{b}$ through another level of parameters μ_a, μ_b, τ_a, and τ_b.

Likelihood, priors and hyperpriors

- Likelihood / data information: $Y_{ij} \sim N(a_i + \beta trt_i + b_i(x_j - \bar{x}), \tau_0^{-1})$.
- Priors: $a_i \sim N(\mu_a, \tau_a^{-1})$, $b_i \sim N(\mu_b, \tau_b^{-1})$, non-informative priors are specified for τ_0 and β: $\tau_0 \sim$ Gamma(ϵ, ϵ), $\beta \sim N(0, 10^6)$.
- Vague hyper-priors: $\mu_a, \mu_b \sim N(0, 10^6)$ and $\tau_a, \tau_b \sim$ Gamma(ϵ, ϵ).
- The fully Bayesian model (2) consists of three levels: data-based likelihood level $p(\mathbf{y}|\boldsymbol{\theta})$, prior level $p(\boldsymbol{\theta}|\boldsymbol{\psi})$, and hyperprior level $p(\boldsymbol{\psi})$.
- Complex models may involve more levels, but models with more than four levels are unusual and unhelpful.
- Information contribution to posterior: Likelihood $>$ prior $>$ hyperprior.

BUGS program

```
model{
  #likelihood p(Y|theta)
for( i in 1 : N ) {
for( j in 1 : T ) {
Y[i , j] ~ dnorm(mu[i , j],tau.0)
mu[i , j] <- a[i] + beta * trt[i] + b[i] * (x[j] - xbar)
}
#Prior p(theta|Psi)
a[i] ~ dnorm(mu.a, tau.a)
b[i] ~ dnorm(mu.b, tau.b)
}
#prior
tau.0 ~ dgamma(0.001,0.001)
beta ~ dnorm(0.0,1.0E-6)
#hyper-priors
mu.a ~ dnorm(0.0,1.0E-6)
mu.b ~ dnorm(0.0,1.0E-6)
tau.a ~ dgamma(0.001,0.001)
tau.b ~ dgamma(0.001,0.001)
#parameters of interest
sigma <- 1 / sqrt(tau.0) #error sd
w0[1] <- mu.a - xbar * mu.b #weight at birth for 1st group
w0[2] <- mu.a + beta - xbar * mu.b #weight at birth for 2nd group
}
```

BUGS data

- List format created from R, but be careful about two issues:
  1. list data obtained from R do not have the required `.Data` keyword for BUGS. Add this keyword for BUGS.
  2. BUGS reads matrix in a different way from R. For example, there is a matrix $M : 5 \times 3$ in R. In order to use it in BUGS, follow this procedure:
     (a) transpose M: `M <- t(M)`;
     (b) dump M: `dput(M, "M.dat")`;
     (c) open `M.dat`, add `.Data` keyword and change `.Dim = c(3,5)` to `.Dim = c(5,3)`.
- List data example

```
list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), xbar = 22, N = 30, T = 5,
Y = structure(
.Data =   c(151, 199, 246, 283, 320,
 145, 199, 249, 293, 354,
 147, 214, 263, 312, 328,
```

BUGS data

- Table format

```
n[] x[]
47   0
148 18
119  8
END
```

Initialize MCMC

- BUGS may automatically generate initial values, but it is highly recommended to provide initial values for fixed effects. Good initial values potentially improve convergence.
- For this model, the fixed effects are $\mu_a, \mu_b, \beta, \tau_0, \tau_a,$ and τ_b. So it is recommended to initialize at least these parameters.
- `list(mu.a = 150, mu.b = 10, beta=0, tau.0 = 1, tau.a = 1, tau.b = 1)`

Procedure to run WinBUGS

See live demonstration.

1. Check code: select `Specification` from the `Model` menu. Highlight `list` in the code, and click `check model` button.
2. Load data: Then highlight `list` in the data code, and click `load data`.
3. Compile: click `compile` button and select the number of MCMC chains.
4. Initialize model: click `initialize` button.
5. Burn-in: Pull down `Model` menu and click `Update`.
6. Monitor samples: click `Samples`. Type parameters of interest and click `set` button.

Procedure to run OpenBUGS using BRUGS package in R

See live demonstration. *The current `BRugs` package only work on Windows.*

1. Create three text files namely `ratsmodel.txt`, `ratsdata.txt`, `ratsinits.txt` and save the three pieces of code in these files, respectively.
2. loading BRugs: `library(BRugs)`
3. Check code: `modelCheck("ratsmodel.txt")`
4. Load data: `modelData("ratsdata.txt")`
5. Compile: `modelCompile(numChains=2)`
6. Initialize model: `modelInits(rep("ratsinits.txt", 2))`
7. Burn-in: `modelUpdate(1000)`
8. Monitor samples: `samplesSet(c("w0", "beta"))`
9. More samples: `modelUpdate(1000)`
10. Statistical inference and plots are also available (see BRugs package information).

Results and interpretation

See live demonstration.

- The posterior densities of these parameters can be estimated by the MCMC samples after convergence.
- Since 95%CI of β covers 0, there is no significant difference between these two groups at .05 level.
- As a conclusion, once we have the distribution of a parameter of interest, we completely know that parameter in statistical sense, so we can do whatever inference from it.