Bayesian Statistics

WinBUGS

Nan Lin
Department of Mathematics
Washington University in St. Louis

Hierarchical model in WinBUGS

- Pluses:
  - Automated Markov chain simulation of the posterior distribution resulting from any user-specified model.
  - No derivation required.
  - Useful for double-checking hand coded results.
  - Great for problems with no exact analytic solution.
  - R2WinBUGS has the computational advantages of BUGS and the statistical and graphical capacities of R.
- Minuses:
  - Black box.
  - Inflexible.

SAT coaching example

- Data

School	y_i	σ_i
A	28.39	14.9
B	7.94	10.2
C	-2.75	16.3
D	6.82	11.0
E	-0.64	9.4
F	0.63	11.4
G	18.01	10.4
H	12.16	17.6

- Model

$$
\begin{aligned}
y_i &\sim N(\theta_i, \sigma_i^2) \\
\theta_i &\sim N(\mu_\theta, \sigma_\theta^2)
\end{aligned}
$$

- Hyperprior

$$
\begin{aligned}
\mu_\theta &\sim N(0, 10^6) \\
\sigma_\theta &\sim U(0, 100)
\end{aligned}
$$

- WinBUGS requires proper priors.
  - So, a flat prior can be expressed through a proper distribution with a large variance.
  - Normal distribution is parametrized by its precision $\tau = 1/\sigma^2$

Bugs code

```
model {
  for (j in 1:J){
    y[j] ~ dnorm(theta[j], tau.y[j])
    theta[j] ~ dnorm(mu.theta, tau.theta)
    tau.y[j] <- pow(sigma.y[j], -2)
  }
  mu.theta ~ dnorm(0, 1.0E-6)
  tau.theta <- pow(sigma.theta, -2)
  sigma.theta ~ dunif(0, 1000)
}
```

Enter the data

```
list(y = c(28,  8, -3,  7, -1,  1, 18, 12),
     sigma.y = c(15, 10, 16, 11,  9, 11, 10, 18),
     J = 8)
```

Initialize the chain

```
list(theta=c(28,  8, -3,  7, -1,  1, 18, 12),
     mu.theta=0, sigma.theta=1)
```

Operation in WinBUGS

- Specify the model

1. Click '`Model`' $\Rightarrow$ '`Specification`'.
   - Highlight '`model`', click '`check model`'.
     - '`model is syntactically correct`'
   - Highlight '`list`' at the beginning of your dataset, click '`load data`'.
     - '`data loaded`'
   - Click '`compile`'.
     - '`model compiled`'
   - Highlight '`list`' at the beginning of your initial values list, click '`load inits`'.
     - '`model is initialized`'
2. Close the '`Model Specification`' window.

Run MCMC in WinBUGS

3. Click ‘`Inference`’ $\Rightarrow$ ‘`Samples`’.
   - In the ‘`node`’ box type ‘`theta`’, click ‘`set`’.
   - In the ‘`node`’ box type ‘`mu.theta`’, click ‘`set`’.
   - In the ‘`node`’ box type ‘`sigma.theta`’, click ‘`set`’.
4. Close the ‘`Sample Monitor Tool`’ window.
5. Click ‘`Model`’ $\Rightarrow$ ‘`Update`’.
   - Click ‘`update`’.
6. Close the ‘`Update Tool`’ window.

Summarize posterior inference

7. Click ‘Inference’ ⇒ ‘Samples’.
   - Choose a parameter of interest from the ‘node’ box.
   - Click ‘history’ for trace plots.
   - Click ‘density’ for marginal posterior densities.
   - Click ‘stats’ for posterior means, quantiles, etc.
8. Click ‘Inference’ ⇒ ‘Compare’.
   - Enter ‘theta’ in the ‘node’ box.
   - Click ‘box plot’ for comparative box plots.
   - Click ‘caterpillar’ for a Lab 4 like plot of the θ_j's.
9. Click ‘Inference’ ⇒ ‘Correlations’.
   - Enter parameter(s) of interest in the ‘node’ box(es).
   - Click ‘scatter’ for scatter plots.
   - Click ‘matrix’ for a matrix-like graph.
   - Click ‘print’ for numerical estimate.

Call WinBUGS in R

- R2WinBUGS is an R package that runs WinBUGS through R.
  - R2WinBUGS has the computational advantages of BUGS and the statistical and graphical capacities of R.
- Install Gelman's applied regression modelling package. Doing so will automatically install the R2WinBUGS package. Then install the the BRugs package:

```
install.packages("arm")
install.packages("BRugs")
library(arm)
library(BRugs)
```

  - BRugs requires R-2.14.2

Enter the data

```
y <- c(28,  8, -3,  7, -1,  1, 18, 12)
J <- length(y)
sigma.y <- c(15, 10, 16, 11,  9, 11, 10, 18)
data <- list ("J", "y", "sigma.y")
```

Initialize the chain

```
inits <- list(theta=y, mu.theta=0, sigma.theta=1)
```

Run MCMC with R2WinBUGS

- Identify parameters to save

```
parameters <- c("theta", "mu.theta", "sigma.theta")
```

- Save the model as 'school.bug.txt'
-
```
library(R2WinBUGS)
schools.sim <- bugs(data, inits, parameters,
                    "schools.bug", n.chains=3,
                    n.iter=1000)
```

  - While BUGS is running, it opens a new window and freezes R.
  - If you run into trouble, add the `` `debug=T' `` option to the `bugs()` command. This will keep the WinBUGS window open after the run so that you can take a look at the log file.

Summarize the posterior

```
> print(schools.sim)

Inference for Bugs model at "schools.bug", fit using WinBUGS,
 3 chains, each with 1000 iterations (first 500 discarded)
 n.sims = 1500 iterations saved

            mean  sd  2.5%  25%  50%  75% 97.5% Rhat n.eff
theta[1]    10.8 7.9  -2.4  5.4 10.2 14.3  29.3  1.0   540
theta[2]     7.8 6.0  -4.1  4.2  7.9 11.2  20.7  1.0   370
theta[3]     5.9 7.5 -11.0  1.8  6.5 10.9  19.8  1.0   290
theta[4]     7.7 6.3  -5.3  4.1  7.8 11.2  21.0  1.0   250
theta[5]     5.1 6.4  -9.2  1.5  5.4 10.2  15.6  1.0    96
theta[6]     6.0 6.7  -8.6  2.4  6.5 10.8  18.4  1.0   220
theta[7]    10.4 6.4  -1.0  5.7 10.4 13.6  24.9  1.0  1200
theta[8]     7.9 7.4  -7.4  4.0  8.1 11.5  23.5  1.0   180
mu.theta     7.7 4.9  -2.1  4.7  7.5 11.0  17.8  1.0   230
sigma.theta  6.2 5.7   0.0  1.9  5.0  8.7  21.2  1.1    80
deviance    60.2 2.2  56.7 59.1 59.9 60.9  65.9  1.0   350

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 2.4 and DIC = 62.6
DIC is an estimate of expected predictive error (lower deviance is better).
```

Deviance: a summary measure of model fit

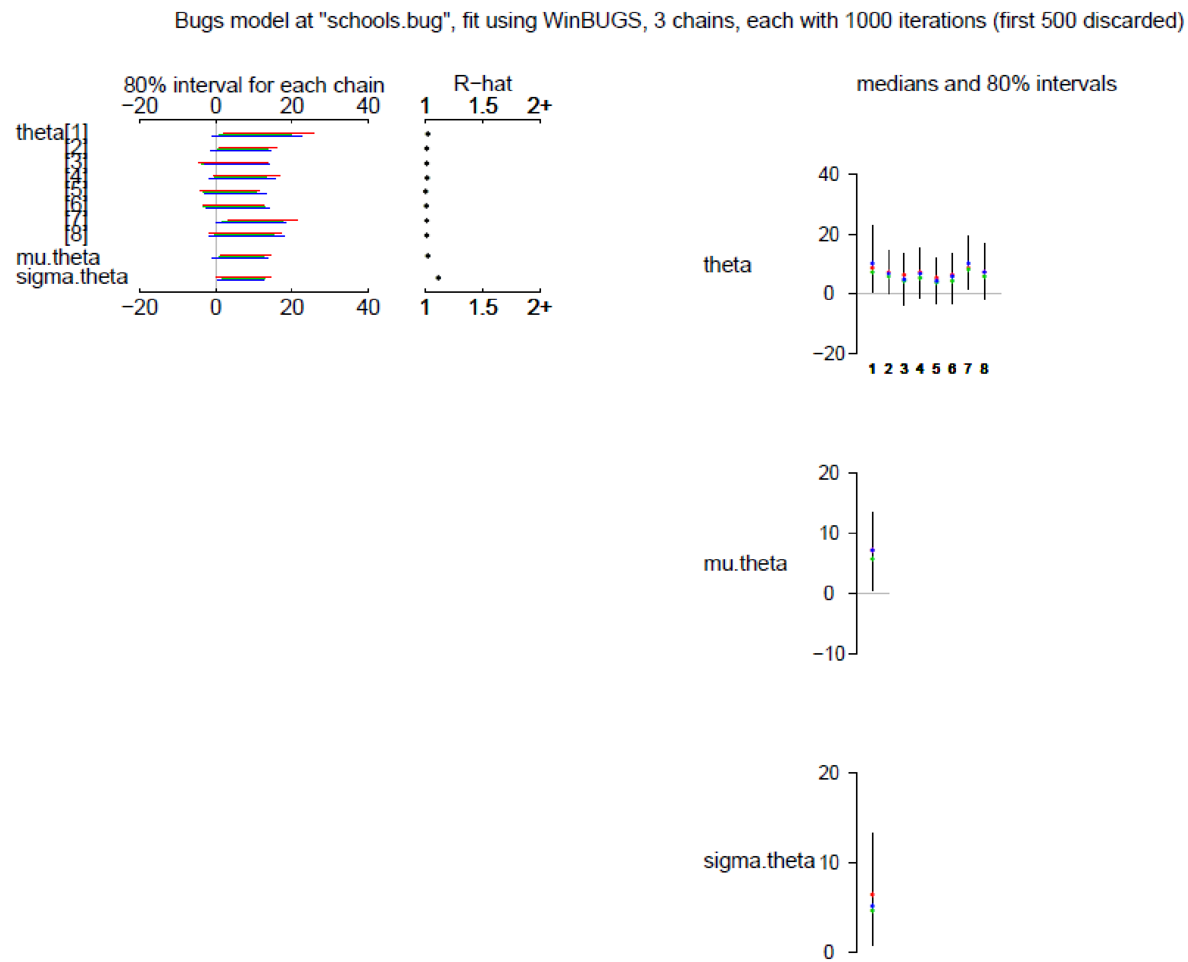
$$D(\theta) = -2 \log L(\theta) + 2h(y),$$

- $L(\theta)$: Likelihood
- $h(y)$: a standardizing function of the data

- If one combines Deviance with an estimate of model complexity, one obtains the deviance information criterion (DIC), which can be used to select models.
  - A model with a smaller DIC is preferred. (Note: this only applies to different models fit based on the same data set.)

Graphical summary of the posterior

```
> plot(schools.sim)
```



More summary of the posterior

- `attach.all(schools.sim$sims.list)`
- This creates:
  - ‘`mu.theta`’: a vector of 1500 simulations of μ_θ.
  - ‘`sigma.theta`’: a vector of 1500 simulations of σ_θ.
  - ‘`theta`’: a matrix of 1500 $\times$ 8 simulations of θ.
- `plot(density(sigma.theta))`
- `mean(mu.theta>0)`
- `cor(theta)`